

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

Network Working Group  
Request for Comments: DRAFT  
Replaces: RFC 788, 780, 772

J. Postel  
ISI  
August 1982

## SIMPLE MAIL TRANSFER PROTOCOL

### 1. INTRODUCTION

The objective of Simple Mail Transfer Protocol (SMTP) is to transfer mail reliably and efficiently.

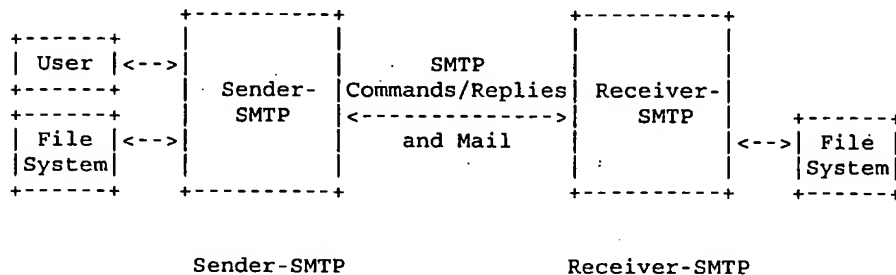
SMTP is independent of the particular transmission subsystem and requires only a reliable ordered data stream channel. Appendices A, B, C, and D describe the use of SMTP with various transport services. A Glossary provides the definitions of terms as used in this document.

An important feature of SMTP is its capability to relay mail across transport service environments. A transport service provides an interprocess communication environment (IPCE). An IPCE may cover one network, several networks, or a subset of a network. It is important to realize that transport systems (or IPCEs) are not one-to-one with networks. A process can communicate directly with another process through any mutually known IPCE. Mail is an application or use of interprocess communication. Mail can be communicated between processes in different IPCEs by relaying through a process connected to two (or more) IPCEs. More specifically, mail can be relayed between hosts on different transport systems by a host on both transport systems.

## 2. THE SMTP MODEL

The SMTP design is based on the following model of communication: as the result of a user mail request, the sender-SMTP establishes a two-way transmission channel to a receiver-SMTP. The receiver-SMTP may be either the ultimate destination or an intermediate. SMTP commands are generated by the sender-SMTP and sent to the receiver-SMTP. SMTP replies are sent from the receiver-SMTP to the sender-SMTP in response to the commands.

Once the transmission channel is established, the SMTP-sender sends a MAIL command indicating the sender of the mail. If the SMTP-receiver can accept mail it responds with an OK reply. The SMTP-sender then sends a RCPT command identifying a recipient of the mail. If the SMTP-receiver can accept mail for that recipient it responds with an OK reply; if not, it responds with a reply rejecting that recipient (but not the whole mail transaction). The SMTP-sender and SMTP-receiver may negotiate several recipients. When the recipients have been negotiated the SMTP-sender sends the mail data, terminating with a special sequence. If the SMTP-receiver successfully processes the mail data it responds with an OK reply. The dialog is purposely lock-step, one-at-a-time.



Model for SMTP Use

Figure 1

The SMTP provides mechanisms for the transmission of mail; directly from the sending user's host to the receiving user's host when the

two host are connected to the same transport service, or via one or more relay SMTP-servers when the source and destination hosts are not connected to the same transport service.

To be able to provide the relay capability the SMTP-server must be supplied with the name of the ultimate destination host as well as the destination mailbox name.

The argument to the MAIL command is a reverse-path, which specifies who the mail is from. The argument to the RCPT command is a forward-path, which specifies who the mail is to. The forward-path is a source route, while the reverse-path is a return route (which may be used to return a message to the sender when an error occurs with a relayed message).

When the same message is sent to multiple recipients the SMTP encourages the transmission of only one copy of the data for all the recipients at the same destination host.

The mail commands and replies have a rigid syntax. Replies also have a numeric code. In the following, examples appear which use actual commands and replies. The complete lists of commands and replies appears in Section 4 on specifications.

Commands and replies are not case sensitive. That is, a command or reply word may be upper case, lower case, or any mixture of upper and lower case. Note that this is not true of mailbox user names. For some hosts the user name is case sensitive, and SMTP implementations must take case to preserve the case of user names as they appear in mailbox arguments. Host names are not case sensitive.

Commands and replies are composed of characters from the ASCII character set [1]. When the transport service provides an 8-bit byte (octet) transmission channel, each 7-bit character is transmitted right justified in an octet with the high order bit cleared to zero.

When specifying the general form of a command or reply, an argument (or special symbol) will be denoted by a meta-linguistic variable (or constant), for example, "<string>" or "<reverse-path>". Here the angle brackets indicate these are meta-linguistic variables. However, some arguments use the angle brackets literally. For example, an actual reverse-path is enclosed in angle brackets, i.e., "<John.Smith@USC-ISI.ARPA>" is an instance of <reverse-path> (the angle brackets are actually transmitted in the command or reply).

### 3. THE SMTP PROCEDURES

This section presents the procedures used in SMTP in several parts. First comes the basic mail procedure defined as a mail transaction. Following this are descriptions of forwarding mail, verifying mailbox names and expanding mailing lists, sending to terminals instead of or in combination with mailboxes, and the opening and closing exchanges. At the end of this section are comments on relaying, a note on mail domains, and a discussion of changing roles. Throughout this section are examples of partial command and reply sequences, several complete scenarios are presented in Appendix F.

#### 3.1. MAIL

There are three steps to SMTP mail transactions. The transaction is started with a MAIL command which gives the sender identification. A series of one or more RCPT commands follows giving the receiver information. Then a DATA command gives the mail data. And finally, the end of mail data indicator confirms the transaction.

The first step in the procedure is the MAIL command. The <reverse-path> contains the source mailbox.

MAIL <SP> FROM:<reverse-path> <CRLF>

This command tells the SMTP-receiver that a new mail transaction is starting and to reset all its state tables and buffers, including any recipients or mail data. It gives the reverse-path which can be used to report errors. If accepted, the receiver-SMTP returns a 250 OK reply.

The <reverse-path> can contain more than just a mailbox. The <reverse-path> is a reverse source routing list of hosts and source mailbox. The first host in the <reverse-path> should be the host sending this command.

The second step in the procedure is the RCPT command.

RCPT <SP> TO:<forward-path> <CRLF>

This command gives a forward-path identifying one recipient. If accepted, the receiver-SMTP returns a 250 OK reply, and stores the forward-path. If the recipient is unknown the receiver-SMTP returns a 550 Failure reply. This second step of the procedure can be repeated any number of times.

The <forward-path> can contain more than just a mailbox. The <forward-path> is a source routing list of hosts and the destination mailbox. The first host in the <forward-path> should be the host receiving this command.

The third step in the procedure is the DATA command.

DATA <CRLF>

If accepted, the receiver-SMTP returns a 354 Intermediate reply and considers all succeeding lines to be the message text. When the end of text is received and stored the SMTP-receiver sends a 250 OK reply.

Since the mail data is sent on the transmission channel the end of the mail data must be indicated so that the command and reply dialog can be resumed. SMTP indicates the end of the mail data by sending a line containing only a period. A transparency procedure is used to prevent this from interfering with the user's text (see Section 4.5.2).

Please note that the mail data includes the memo header items such as Date, Subject, To, Cc, From [2].

The end of mail data indicator also confirms the mail transaction and tells the receiver-SMTP to now process the stored recipients and mail data. If accepted, the receiver-SMTP returns a 250 OK reply. The DATA command should fail only if the mail transaction was incomplete (for example, no recipients), or if resources are not available.

The above procedure is an example of a mail transaction. These commands must be used only in the order discussed above. Example 1 (below) illustrates the use of these commands in a mail transaction.

---

Example of the SMTP Procedure

This SMTP example shows mail sent by Smith at host Alpha.ARPA, to Jones, Green, and Brown at host Beta.ARPA. Here we assume that host Alpha contacts host Beta directly.

S: MAIL FROM:<Smith@Alpha.ARPA>  
R: 250 OK

S: RCPT TO:<Jones@Beta.ARPA>  
R: 250 OK

S: RCPT TO:<Green@Beta.ARPA>  
R: 550 No such user here

S: RCPT TO:<Brown@Beta.ARPA>  
R: 250 OK

S: DATA  
R: 354 Start mail input; end with <CRLF>.<CRLF>  
S: Blah blah blah...  
S: ...etc. etc. etc.  
S: <CRLF>.<CRLF>  
R: 250 OK

The mail has now been accepted for Jones and Brown. Green did not have a mailbox at host Beta.

Example 1

---

## 3.2. FORWARDING

There are some cases where the destination information in the <forward-path> is incorrect, but the receiver-SMTP knows the correct destination. In such cases, one of the following replies should be used to allow the sender to contact the correct destination.

251 User not local; will forward to <forward-path>

This reply indicates that the receiver-SMTP knows the user's mailbox is on another host and indicates the correct forward-path to use in the future. Note that either the host or user or both may be different. The receiver takes responsibility for delivering the message.

551 User not local; please try <forward-path>

This reply indicates that the receiver-SMTP knows the user's mailbox is on another host and indicates the correct forward-path to use. Note that either the host or user or both may be different. The receiver refuses to accept mail for this user, and the sender must either redirect the mail according to the information provided or return an error response to the originating user.

Example 2 illustrates the use of these responses.

---

Example of Forwarding

Either

S: RCPT TO:<Postel@USC-ISI.ARPA>  
R: 251 User not local; will forward to <Postel@USC-ISIF.ARPA>

Or

S: RCPT TO:<Paul@USC-ISIB.ARPA>  
R: 551 User not local; please try <Mockapetris@USC-ISIF.ARPA>

Example 2

---



### 3.3. VERIFYING AND EXPANDING

SMTP provides as additional features, commands to verify a user name or expand a mailing list. This is done with the VRFY and EXPN commands, which have character string arguments. For the VRFY command, the string is a user name, and the response may include the full name of the user and must include the mailbox of the user. For the EXPN command, the string identifies a mailing list, and the multiline response may include the full name of the users and must give the mailboxes on the mailing list.

"User name" is a fuzzy term and used purposely. If a host implements the VRFY or EXPN commands then at least local mailboxes must be recognized as "user names". If a host chooses to recognize other strings as "user names" that is allowed.

In some hosts the distinction between a mailing list and an alias for a single mailbox is a bit fuzzy, since a common data structure may hold both types of entries, and it is possible to have mailing lists of one mailbox. If a request is made to verify a mailing list a positive response can be given if on receipt of a message so addressed it will be delivered to everyone on the list, otherwise an error should be reported (e.g., "550 That is a mailing list, not a user"). If a request is made to expand a user name a positive response can be formed by returning a list containing one name, or an error can be reported (e.g., "550 That is a user name, not a mailing list").

In the case of a multiline reply (normal for EXPN) exactly one mailbox is to be specified on each line of the reply. In the case of an ambiguous request, for example, "VRFY Smith", where there are two Smith's the response must be "553 User ambiguous".

The case of verifying a user name is straightforward as shown in example 3.

---

Example of Verifying a User Name

Either

S: VRFY Smith  
R: 250 Fred Smith <Smith@USC-ISIF.ARPA>

Or

S: VRFY Smith  
R: 251 User not local; will forward to <Smith@USC-ISIQ.ARPA>

Or

S: VRFY Jones  
R: 550 String does not match anything.

Or

S: VRFY Jones  
R: 551 User not local; please try <Jones@USC-ISIQ.ARPA>

Or

S: VRFY Gourzenkyinplatz  
R: 553 User ambiguous.

Example 3

---

The case of expanding a mailbox list requires a multiline reply as shown in example 4.

---

Example of Expanding a Mailing List

Either

```
S: EXPN Example-People
R: 250-Jon Postel <Postel@USC-ISIF.ARPA>
R: 250-Fred Fonebone <Fonebone@USC-ISIQ.ARPA>
R: 250-Sam Q. Smith <SQSmith@USC-ISIQ.ARPA>
R: 250-Quincy Smith <@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>
R: 250-<joe@foo-unix.ARPA>
R: 250 <xyz@bar-unix.ARPA>
```

Or

```
S: EXPN Executive-Washroom-List
R: 550 Access Denied to You.
```

Example 4

---

The character string arguments of the VRFY and EXPN commands cannot be further restricted due to the variety of implementations of the user name and mailbox list concepts. On some systems it may be appropriate for the argument of the EXPN command to be a file name for a file containing a mailing list, but again there is a variety of file naming conventions in the Internet.

The VRFY and EXPN commands are not included in the minimum implementation (Section 4.5.1), and are not required to work across relays when they are implemented.

### 3.4. SENDING AND MAILING

The main purpose of SMTP is to deliver messages to user's mailboxes. A very similar service provided by some hosts is to deliver messages to user's terminals (provided the user is active on the host). The delivery to the user's mailbox is called "mailing", the delivery to the user's terminal is called "sending". Because in many hosts the implementation of sending is nearly identical to the implementation of mailing these two functions are combined in SMTP. However the sending commands are not included in the required minimum implementation (Section 4.5.1). Users should have the ability to control the writing of messages on their terminals. Most hosts permit the users to accept or refuse such messages.

The following three command are defined to support the sending options. These are used in the mail transaction instead of the MAIL command and inform the receiver-SMTP of the special semantics of this transaction:

SEND <SP> FROM:<reverse-path> <CRLF>

The SEND command requires that the mail data be delivered to the user's terminal. If the user is not active (or not accepting terminal messages) on the host a 450 reply may be returned to a RCPT command. The mail transaction is successful if the message is delivered to the terminal.

SOML <SP> FROM:<reverse-path> <CRLF>

The Send Or Mail command requires that the mail data be delivered to the user's terminal if the user is active (and accepting terminal messages) on the host. If the user is not active (or not accepting terminal messages) then the mail data is entered into the user's mailbox. The mail transaction is successful if the message is delivered either to the terminal or the mailbox.

SAML <SP> FROM:<reverse-path> <CRLF>

The Send And Mail command requires that the mail data be delivered to the user's terminal if the user is active (and accepting terminal messages) on the host. In any case the mail data is entered into the user's mailbox. The mail transaction is successful if the message is delivered to the mailbox.

August 1982  
Simple Mail Transfer Protocol

RFC 821

The same reply codes that are used for the MAIL commands are used for these commands.

## 3.5. OPENING AND CLOSING

At the time the transmission channel is opened there is an exchange to ensure that the hosts are communicating with the hosts they think they are.

The following two commands are used in transmission channel opening and closing:

HELO <SP> <domain> <CRLF>

QUIT <CRLF>

In the HELO command the host sending the command identifies itself; the command may be interpreted as saying "Hello, I am <domain>".

-----  
Example of Connection Opening

R: 220 BBN-UNIX.ARPA Simple Mail Transfer Service Ready  
S: HELO USC-ISIF.ARPA  
R: 250 BBN-UNIX.ARPA

Example 5  
-----  
-----

## Example of Connection Closing

S: QUIT  
R: 221 BBN-UNIX.ARPA Service closing transmission channel

Example 6  
-----

### 3.6. RELAYING

The forward-path may be a source route of the form "@ONE,@TWO:JOE@THREE", where ONE, TWO, and THREE are hosts. This form is used to emphasize the distinction between an address and a route. The mailbox is an absolute address, and the route is information about how to get there. The two concepts should not be confused.

Conceptually the elements of the forward-path are moved to the reverse-path as the message is relayed from one server-SMTP to another. The reverse-path is a reverse source route, (i.e., a source route from the current location of the message to the originator of the message). When a server-SMTP deletes its identifier from the forward-path and inserts it into the reverse-path, it must use the name it is known by in the environment it is sending into, not the environment the mail came from, in case the server-SMTP is known by different names in different environments.

If when the message arrives at an SMTP the first element of the forward-path is not the identifier of that SMTP the element is not deleted from the forward-path and is used to determine the next SMTP to send the message to. In any case, the SMTP adds its own identifier to the reverse-path.

Using source routing the receiver-SMTP receives mail to be relayed to another server-SMTP. The receiver-SMTP may accept or reject the task of relaying the mail in the same way it accepts or rejects mail for a local user. The receiver-SMTP transforms the command arguments by moving its own identifier from the forward-path to the beginning of the reverse-path. The receiver-SMTP then becomes a sender-SMTP, establishes a transmission channel to the next SMTP in the forward-path, and sends it the mail.

The first host in the reverse-path should be the host sending the SMTP commands, and the first host in the forward-path should be the host receiving the SMTP commands.

Notice that the forward-path and reverse-path appear in the SMTP commands and replies, but not necessarily in the message. That is, there is no need for these paths and especially this syntax to appear in the "To:", "From:", "CC:", etc. fields of the message header.

If a server-SMTP has accepted the task of relaying the mail and

later finds that the forward-path is incorrect or that the mail cannot be delivered for whatever reason, then it must construct an "undeliverable mail" notification message and send it to the originator of the undeliverable mail (as indicated by the reverse-path).

This notification message must be from the server-SMTP at this host. Of course, server-SMTPs should not send notification messages about problems with notification messages. One way to prevent loops in error reporting is to specify a null reverse-path in the MAIL command of a notification message. When such a message is relayed it is permissible to leave the reverse-path null. A MAIL command with a null reverse-path appears as follows:

MAIL FROM:<>

An undeliverable mail notification message is shown in example 7. This notification is in response to a message originated by JOE at HOSTW and sent via HOSTX to HOSTY with instructions to relay it on to HOSTZ. What we see in the example is the transaction between HOSTY and HOSTX, which is the first step in the return of the notification message.



---

Example Undeliverable Mail Notification Message

S: MAIL FROM:<>  
R: 250 ok  
S: RCPT TO:<@HOSTX.ARPA:JOE@HOSTW.ARPA>  
R: 250 ok  
S: DATA  
R: 354 send the mail data, end with .  
S: Date: 23 Oct 81 11:22:33  
S: From: SMTP@HOSTY.ARPA  
S: To: JOE@HOSTW.ARPA  
S: Subject: Mail System Problem  
S:  
S:   Sorry JOE, your message to SAM@HOSTZ.ARPA lost.  
S:   HOSTZ.ARPA said this:  
S:   "550 No Such User"  
S: .  
R: 250 ok

Example 7

---

### 3.7. DOMAINS

Domains are a recently introduced concept in the ARPA Internet mail system. The use of domains changes the address space from a flat global space of simple character string host names to a hierarchically structured rooted tree of global addresses. The host name is replaced by a domain and host designator which is a sequence of domain element strings separated by periods with the understanding that the domain elements are ordered from the most specific to the most general.

For example, "USC-ISIF.ARPA", "Fred.Cambridge.UK", and "PC7.LCS.MIT.ARPA" might be host-and-domain identifiers.

Whenever domain names are used in SMTP only the official names are used, the use of nicknames or aliases is not allowed.

### 3.8. CHANGING ROLES

The TURN command may be used to reverse the roles of the two programs communicating over the transmission channel.

If program-A is currently the sender-SMTP and it sends the TURN command and receives an ok reply (250) then program-A becomes the receiver-SMTP.

If program-B is currently the receiver-SMTP and it receives the TURN command and sends an ok reply (250) then program-B becomes the sender-SMTP.

To refuse to change roles the receiver sends the 502 reply.

Please note that this command is optional. It would not normally be used in situations where the transmission channel is TCP. However, when the cost of establishing the transmission channel is high, this command may be quite useful. For example, this command may be useful in supporting be mail exchange using the public switched telephone system as a transmission channel, especially if some hosts poll other hosts for mail exchanges.

#### 4. THE SMTP SPECIFICATIONS

##### 4.1. SMTP COMMANDS

###### 4.1.1. COMMAND SEMANTICS

The SMTP commands define the mail transfer or the mail system function requested by the user. SMTP commands are character strings terminated by <CRLF>. The command codes themselves are alphabetic characters terminated by <SP> if parameters follow and <CRLF> otherwise. The syntax of mailboxes must conform to receiver site conventions. The SMTP commands are discussed below. The SMTP replies are discussed in the Section 4.2.

A mail transaction involves several data objects which are communicated as arguments to different commands. The reverse-path is the argument of the MAIL command, the forward-path is the argument of the RCPT command, and the mail data is the argument of the DATA command. These arguments or data objects must be transmitted and held pending the confirmation communicated by the end of mail data indication which finalizes the transaction. The model for this is that distinct buffers are provided to hold the types of data objects, that is, there is a reverse-path buffer, a forward-path buffer, and a mail data buffer. Specific commands cause information to be appended to a specific buffer, or cause one or more buffers to be cleared.

###### HELLO (HELO)

This command is used to identify the sender-SMTP to the receiver-SMTP. The argument field contains the host name of the sender-SMTP.

The receiver-SMTP identifies itself to the sender-SMTP in the connection greeting reply, and in the response to this command.

This command and an OK reply to it confirm that both the sender-SMTP and the receiver-SMTP are in the initial state, that is, there is no transaction in progress and all state tables and buffers are cleared.

#### MAIL (MAIL)

This command is used to initiate a mail transaction in which the mail data is delivered to one or more mailboxes. The argument field contains a reverse-path.

The reverse-path consists of an optional list of hosts and the sender mailbox. When the list of hosts is present, it is a "reverse" source route and indicates that the mail was relayed through each host on the list (the first host in the list was the most recent relay). This list is used as a source route to return non-delivery notices to the sender. As each relay host adds itself to the beginning of the list, it must use its name as known in the IPCE to which it is relaying the mail rather than the IPCE from which the mail came (if they are different). In some types of error reporting messages (for example, undeliverable mail notifications) the reverse-path may be null (see Example 7).

This command clears the reverse-path buffer, the forward-path buffer, and the mail data buffer; and inserts the reverse-path information from this command into the reverse-path buffer.

#### RECIPIENT (RCPT)

This command is used to identify an individual recipient of the mail data; multiple recipients are specified by multiple use of this command.

The forward-path consists of an optional list of hosts and a required destination mailbox. When the list of hosts is present, it is a source route and indicates that the mail must be relayed to the next host on the list. If the receiver-SMTP does not implement the relay function it may use the same reply it would for an unknown local user (550).

When mail is relayed, the relay host must remove itself from the beginning forward-path and put itself at the beginning of the reverse-path. When mail reaches its ultimate destination (the forward-path contains only a destination mailbox), the receiver-SMTP inserts it into the destination mailbox in accordance with its host mail conventions.

For example, mail received at relay host A with arguments

```
FROM:<USERX@HOSTY.ARPA>  
TO:<@HOSTA.ARPA,@HOSTB.ARPA:USERC@HOSTD.ARPA>
```

will be relayed on to host B with arguments

```
FROM:<@HOSTA.ARPA:USERX@HOSTY.ARPA>  
TO:<@HOSTB.ARPA:USERC@HOSTD.ARPA>.
```

This command causes its forward-path argument to be appended to the forward-path buffer.

#### DATA (DATA)

The receiver treats the lines following the command as mail data from the sender. This command causes the mail data from this command to be appended to the mail data buffer. The mail data may contain any of the 128 ASCII character codes.

The mail data is terminated by a line containing only a period, that is the character sequence "<CRLF>.<CRLF>" (see Section 4.5.2 on Transparency). This is the end of mail data indication.

The end of mail data indication requires that the receiver must now process the stored mail transaction information. This processing consumes the information in the reverse-path buffer, the forward-path buffer, and the mail data buffer, and on the completion of this command these buffers are cleared. If the processing is successful the receiver must send an OK reply. If the processing fails completely the receiver must send a failure reply.

When the receiver-SMTP accepts a message either for relaying or for final delivery it inserts at the beginning of the mail data a time stamp line. The time stamp line indicates the identity of the host that sent the message, and the identity of the host that received the message (and is inserting this time stamp), and the date and time the message was received. Relayed messages will have multiple time stamp lines.

When the receiver-SMTP makes the "final delivery" of a message it inserts at the beginning of the mail data a

return path line. The return path line preserves the information in the <reverse-path> from the MAIL command. Here, final delivery means the message leaves the SMTP world. Normally, this would mean it has been delivered to the destination user, but in some cases it may be further processed and transmitted by another mail system.

It is possible for the mailbox in the return path be different from the actual sender's mailbox, for example, if error responses are to be delivered a special error handling mailbox rather than the message senders.

The preceding two paragraphs imply that the final mail data will begin with a return path line, followed by one or more time stamp lines. These lines will be followed by the mail data header and body [2]. See Example 8.

Special mention is needed of the response and further action required when the processing following the end of mail data indication is partially successful. This could arise if after accepting several recipients and the mail data, the receiver-SMTP finds that the mail data can be successfully delivered to some of the recipients, but it cannot be to others (for example, due to mailbox space allocation problems). In such a situation, the response to the DATA command must be an OK reply. But, the receiver-SMTP must compose and send an "undeliverable mail" notification message to the originator of the message. Either a single notification which lists all of the recipients that failed to get the message, or separate notification messages must be sent for each failed recipient (see Example 7). All undeliverable mail notification messages are sent using the MAIL command (even if they result from processing a SEND, SOML, or SAML command).

---

Example of Return Path and Received Time Stamps

Return-Path: <@GHI.ARPA,@DEF.ARPA,@ABC.ARPA:JOE@ABC.ARPA>  
Received: from GHI.ARPA by JKL.ARPA ; 27 Oct 81 15:27:39 PST  
Received: from DEF.ARPA by GHI.ARPA ; 27 Oct 81 15:15:13 PST  
Received: from ABC.ARPA by DEF.ARPA ; 27 Oct 81 15:01:59 PST  
Date: 27 Oct 81 15:01:01 PST  
From: JOE@ABC.ARPA  
Subject: Improved Mailing System Installed  
To: SAM@JKL.ARPA

This is to inform you that ...

---

Example 8

---

SEND (SEND)

This command is used to initiate a mail transaction in which the mail data is delivered to one or more terminals. The argument field contains a reverse-path. This command is successful if the message is delivered to a terminal.

The reverse-path consists of an optional list of hosts and the sender mailbox. When the list of hosts is present, it is a "reverse" source route and indicates that the mail was relayed through each host on the list (the first host in the list was the most recent relay). This list is used as a source route to return non-delivery notices to the sender. As each relay host adds itself to the beginning of the list, it must use its name as known in the IPCE to which it is relaying the mail rather than the IPCE from which the mail came (if they are different).

This command clears the reverse-path buffer, the forward-path buffer, and the mail data buffer; and inserts the reverse-path information from this command into the reverse-path buffer.

## SEND OR MAIL (SOML)

This command is used to initiate a mail transaction in which the mail data is delivered to one or more terminals or



mailboxes. For each recipient the mail data is delivered to the recipient's terminal if the recipient is active on the host (and accepting terminal messages), otherwise to the recipient's mailbox. The argument field contains a reverse-path. This command is successful if the message is delivered to a terminal or the mailbox.

The reverse-path consists of an optional list of hosts and the sender mailbox. When the list of hosts is present, it is a "reverse" source route and indicates that the mail was relayed through each host on the list (the first host in the list was the most recent relay). This list is used as a source route to return non-delivery notices to the sender. As each relay host adds itself to the beginning of the list, it must use its name as known in the IPCE to which it is relaying the mail rather than the IPCE from which the mail came (if they are different).

This command clears the reverse-path buffer, the forward-path buffer, and the mail data buffer; and inserts the reverse-path information from this command into the reverse-path buffer.

#### SEND AND MAIL (SAML)

This command is used to initiate a mail transaction in which the mail data is delivered to one or more terminals and mailboxes. For each recipient the mail data is delivered to the recipient's terminal if the recipient is active on the host (and accepting terminal messages), and for all recipients to the recipient's mailbox. The argument field contains a reverse-path. This command is successful if the message is delivered to the mailbox.

The reverse-path consists of an optional list of hosts and the sender mailbox. When the list of hosts is present, it is a "reverse" source route and indicates that the mail was relayed through each host on the list (the first host in the list was the most recent relay). This list is used as a source route to return non-delivery notices to the sender. As each relay host adds itself to the beginning of the list, it must use its name as known in the IPCE to which it is relaying the mail rather than the IPCE from which the mail came (if they are different).

This command clears the reverse-path buffer, the

forward-path buffer, and the mail data buffer; and inserts the reverse-path information from this command into the reverse-path buffer.

#### RESET (RSET)

This command specifies that the current mail transaction is to be aborted. Any stored sender, recipients, and mail data must be discarded, and all buffers and state tables cleared. The receiver must send an OK reply.

#### VERIFY (VRFY)

This command asks the receiver to confirm that the argument identifies a user. If it is a user name, the full name of the user (if known) and the fully specified mailbox are returned.

This command has no effect on any of the reverse-path buffer, the forward-path buffer, or the mail data buffer.

#### EXPAND (EXPN)

This command asks the receiver to confirm that the argument identifies a mailing list, and if so, to return the membership of that list. The full name of the users (if known) and the fully specified mailboxes are returned in a multiline reply.

This command has no effect on any of the reverse-path buffer, the forward-path buffer, or the mail data buffer.

#### HELP (HELP)

This command causes the receiver to send helpful information to the sender of the HELP command. The command may take an argument (e.g., any command name) and return more specific information as a response.

This command has no effect on any of the reverse-path buffer, the forward-path buffer, or the mail data buffer.

NOOP (NOOP)

This command does not affect any parameters or previously entered commands. It specifies no action other than that the receiver send an OK reply.

This command has no effect on any of the reverse-path buffer, the forward-path buffer, or the mail data buffer.

QUIT (QUIT)

This command specifies that the receiver must send an OK reply, and then close the transmission channel.

The receiver should not close the transmission channel until it receives and replies to a QUIT command (even if there was an error). The sender should not close the transmission channel until it send a QUIT command and receives the reply (even if there was an error response to a previous command). If the connection is closed prematurely the receiver should act as if a RSET command had been received (canceling any pending transaction, but not undoing any previously completed transaction), the sender should act as if the command or transaction in progress had received a temporary error (4xx).

TURN (TURN)

This command specifies that the receiver must either (1) send an OK reply and then take on the role of the sender-SMTP, or (2) send a refusal reply and retain the role of the receiver-SMTP.

If program-A is currently the sender-SMTP and it sends the TURN command and receives an OK reply (250) then program-A becomes the receiver-SMTP. Program-A is then in the initial state as if the transmission channel just opened, and it then sends the 220 service ready greeting.

If program-B is currently the receiver-SMTP and it receives the TURN command and sends an OK reply (250) then program-B becomes the sender-SMTP. Program-B is then in the initial state as if the transmission channel just opened, and it then expects to receive the 220 service ready greeting.

To refuse to change roles the receiver sends the 502 reply.

There are restrictions on the order in which these command may be used.

The first command in a session must be the HELO command. The HELO command may be used later in a session as well. If the HELO command argument is not acceptable a 501 failure reply must be returned and the receiver-SMTP must stay in the same state.

The NOOP, HELP, EXPN, and VRFY commands can be used at any time during a session.

The MAIL, SEND, SOML, or SAML commands begin a mail transaction. Once started a mail transaction consists of one of the transaction beginning commands, one or more RCPT commands, and a DATA command, in that order. A mail transaction may be aborted by the RSET command. There may be zero or more transactions in a session.

If the transaction beginning command argument is not acceptable a 501 failure reply must be returned and the receiver-SMTP must stay in the same state. If the commands in a transaction are out of order a 503 failure reply must be returned and the receiver-SMTP must stay in the same state.

The last command in a session must be the QUIT command. The QUIT command can not be used at any other time in a session.

#### 4.1.2. COMMAND SYNTAX

The commands consist of a command code followed by an argument field. Command codes are four alphabetic characters. Upper and lower case alphabetic characters are to be treated identically. Thus, any of the following may represent the mail command:

MAIL Mail mail MaIl mAIl

This also applies to any symbols representing parameter values, such as "TO" or "to" for the forward-path. Command codes and the argument fields are separated by one or more spaces. However, within the reverse-path and forward-path arguments case is important. In particular, in some hosts the user "smith" is different from the user "Smith".

The argument field consists of a variable length character string ending with the character sequence <CRLF>. The receiver is to take no action until this sequence is received.

Square brackets denote an optional argument field. If the option is not taken, the appropriate default is implied.

The following are the SMTP commands:

HELO <SP> <domain> <CRLF>  
MAIL <SP> FROM:<reverse-path> <CRLF>  
RCPT <SP> TO:<forward-path> <CRLF>  
DATA <CRLF>  
RSET <CRLF>  
SEND <SP> FROM:<reverse-path> <CRLF>  
SOML <SP> FROM:<reverse-path> <CRLF>  
SAML <SP> FROM:<reverse-path> <CRLF>  
VRFY <SP> <string> <CRLF>  
EXPN <SP> <string> <CRLF>  
HELP [<SP> <string>] <CRLF>  
NOOP <CRLF>  
QUIT <CRLF>  
TURN <CRLF>

The syntax of the above argument fields (using BNF notation where applicable) is given below. The "..." notation indicates that a field may be repeated one or more times.

```
<reverse-path> ::= <path>

<forward-path> ::= <path>

<path> ::= "<" [ <a-d-l> ":" ] <mailbox> ">"

<a-d-l> ::= <at-domain> | <at-domain> "," <a-d-l>

<at-domain> ::= "@" <domain>

<domain> ::= <element> | <element> "." <domain>

<element> ::= <name> | "#" <number> | "[" <dotnum> "]"

<mailbox> ::= <local-part> "@" <domain>

<local-part> ::= <dot-string> | <quoted-string>

<name> ::= <a> <ldh-str> <let-dig>

<ldh-str> ::= <let-dig-hyp> | <let-dig-hyp> <ldh-str>

<let-dig> ::= <a> | <d>

<let-dig-hyp> ::= <a> | <d> | "-"

<dot-string> ::= <string> | <string> "." <dot-string>

<string> ::= <char> | <char> <string>

<quoted-string> ::= "\"" <qtext> "\""

<qtext> ::= "\" <x> | "\" <x> <qtext> | <q> | <q> <qtext>

<char> ::= <c> | "\" <x>

<dotnum> ::= <snum> "." <snum> "." <snum> "." <snum>

<number> ::= <d> | <d> <number>

<CRLF> ::= <CR> <LF>
```

<CR> ::= the carriage return character (ASCII code 13)  
 <LF> ::= the line feed character (ASCII code 10)  
 <SP> ::= the space character (ASCII code 32)  
 <snum> ::= one, two, or three digits representing a decimal integer value in the range 0 through 255  
 <a> ::= any one of the 52 alphabetic characters A through Z in upper case and a through z in lower case  
 <c> ::= any one of the 128 ASCII characters, but not any <special> or <SP>  
 <d> ::= any one of the ten digits 0 through 9  
 <q> ::= any one of the 128 ASCII characters except <CR>, <LF>, quote ("), or backslash (\)  
 <x> ::= any one of the 128 ASCII characters (no exceptions)  
 <special> ::= "<" | ">" | "(" | ")" | "[" | "]" | "\" | "." | "," | ";" | ":" | "@" | "" | the control characters (ASCII codes 0 through 31 inclusive and 127)

Note that the backslash, "\", is a quote character, which is used to indicate that the next character is to be used literally (instead of its normal interpretation). For example, "Joe\,Smith" could be used to indicate a single nine character user field with comma being the fourth character of the field.

Hosts are generally known by names which are translated to addresses in each host. Note that the name elements of domains are the official names -- no use of nicknames or aliases is allowed.

Sometimes a host is not known to the translation function and communication is blocked. To bypass this barrier two numeric forms are also allowed for host "names". One form is a decimal integer prefixed by a pound sign, "#", which indicates the number is the address of the host. Another form is four small decimal integers separated by dots and enclosed by brackets, e.g., "[123.255.37.2]", which indicates a 32-bit ARPA Internet Address in four 8-bit fields.



The time stamp line and the return path line are formally defined as follows:

<return-path-line> ::= "Return-Path:" <SP><reverse-path><CRLF>

<time-stamp-line> ::= "Received:" <SP> <stamp> <CRLF>

<stamp> ::= <from-domain> <by-domain> <opt-info> ";"  
          <daytime>

<from-domain> ::= "FROM" <SP> <domain> <SP>

<by-domain> ::= "BY" <SP> <domain> <SP>

<opt-info> ::= [<via>] [<with>] [<id>] [<for>]

<via> ::= "VIA" <SP> <link> <SP>

<with> ::= "WITH" <SP> <protocol> <SP>

<id> ::= "ID" <SP> <string> <SP>

<for> ::= "FOR" <SP> <path> <SP>

<link> ::= The standard names for links are registered with  
          the Network Information Center.

<protocol> ::= The standard names for protocols are  
              registered with the Network Information Center.

<daytime> ::= <SP> <date> <SP> <time>

<date> ::= <dd> <SP> <mon> <SP> <yy>

<time> ::= <hh> ":" <mm> ":" <ss> <SP> <zone>

<dd> ::= the one or two decimal integer day of the month in  
          the range 1 to 31.

<mon> ::= "JAN" | "FEB" | "MAR" | "APR" | "MAY" | "JUN" |  
          "JUL" | "AUG" | "SEP" | "OCT" | "NOV" | "DEC"

<yy> ::= the two decimal integer year of the century in the  
          range 00 to 99.

<hh> ::= the two decimal integer hour of the day in the  
range 00 to 24.

<mm> ::= the two decimal integer minute of the hour in the  
range 00 to 59.

<ss> ::= the two decimal integer second of the minute in the  
range 00 to 59.

<zone> ::= "UT" for Universal Time (the default) or other  
time zone designator (as in [2]).

---

Return Path Example

Return-Path: <@CHARLIE.ARPA,@BAKER.ARPA:JOE@ABLE.ARPA>

Example 9

---

---

Time Stamp Line Example

Received: FROM ABC.ARPA BY XYZ.ARPA ; 22 OCT 81 09:23:59 PDT

Received: from ABC.ARPA by XYZ.ARPA via TELENET with X25  
id M12345 for Smith@PDQ.ARPA ; 22 OCT 81 09:23:59 PDT

Example 10

---

#### 4.2. SMTP REPLIES

Replies to SMTP commands are devised to ensure the synchronization of requests and actions in the process of mail transfer, and to guarantee that the sender-SMTP always knows the state of the receiver-SMTP. Every command must generate exactly one reply.

The details of the command-reply sequence are made explicit in Section 5.3 on Sequencing and Section 5.4 State Diagrams.

An SMTP reply consists of a three digit number (transmitted as three alphanumeric characters) followed by some text. The number is intended for use by automata to determine what state to enter next; the text is meant for the human user. It is intended that the three digits contain enough encoded information that the sender-SMTP need not examine the text and may either discard it or pass it on to the user, as appropriate. In particular, the text may be receiver-dependent and context dependent, so there are likely to be varying texts for each reply code. A discussion of the theory of reply codes is given in Appendix E. Formally, a reply is defined to be the sequence: a three-digit code, <SP>, one line of text, and <CRLF>, or a multiline reply (as defined in Appendix E). Only the EXPN and HELP commands are expected to result in multiline replies in normal circumstances, however multiline replies are allowed for any command.

## 4.2.1. REPLY CODES BY FUNCTION GROUPS

500 Syntax error, command unrecognized  
[This may include errors such as command line too long]  
501 Syntax error in parameters or arguments  
502 Command not implemented  
503 Bad sequence of commands  
504 Command parameter not implemented

211 System status, or system help reply  
214 Help message  
[Information on how to use the receiver or the meaning of a particular non-standard command; this reply is useful only to the human user]

220 <domain> Service ready  
221 <domain> Service closing transmission channel  
421 <domain> Service not available,  
closing transmission channel  
[This may be a reply to any command if the service knows it must shut down]

250 Requested mail action okay, completed  
251 User not local; will forward to <forward-path>  
450 Requested mail action not taken: mailbox unavailable  
[E.g., mailbox busy]  
550 Requested action not taken: mailbox unavailable  
[E.g., mailbox not found, no access]  
451 Requested action aborted: error in processing  
551 User not local; please try <forward-path>  
452 Requested action not taken: insufficient system storage  
552 Requested mail action aborted: exceeded storage allocation  
553 Requested action not taken: mailbox name not allowed  
[E.g., mailbox syntax incorrect]  
354 Start mail input; end with <CRLF>.<CRLF>  
554 Transaction failed

4.2.2. NUMERIC ORDER LIST OF REPLY CODES

211 System status, or system help reply  
214 Help message  
    [Information on how to use the receiver or the meaning of a  
    particular non-standard command; this reply is useful only  
    to the human user]  
220 <domain> Service ready  
221 <domain> Service closing transmission channel  
250 Requested mail action okay, completed  
251 User not local; will forward to <forward-path>  
  
354 Start mail input; end with <CRLF>.<CRLF>  
  
421 <domain> Service not available,  
    closing transmission channel  
    [This may be a reply to any command if the service knows it  
    must shut down]  
450 Requested mail action not taken: mailbox unavailable  
    [E.g., mailbox busy]  
451 Requested action aborted: local error in processing  
452 Requested action not taken: insufficient system storage  
  
500 Syntax error, command unrecognized  
    [This may include errors such as command line too long]  
501 Syntax error in parameters or arguments  
502 Command not implemented  
503 Bad sequence of commands  
504 Command parameter not implemented  
550 Requested action not taken: mailbox unavailable  
    [E.g., mailbox not found, no access]  
551 User not local; please try <forward-path>  
552 Requested mail action aborted: exceeded storage allocation  
553 Requested action not taken: mailbox name not allowed  
    [E.g., mailbox syntax incorrect]  
554 Transaction failed

## 4.3. SEQUENCING OF COMMANDS AND REPLIES

The communication between the sender and receiver is intended to be an alternating dialogue, controlled by the sender. As such, the sender issues a command and the receiver responds with a reply. The sender must wait for this response before sending further commands.

One important reply is the connection greeting. Normally, a receiver will send a 220 "Service ready" reply when the connection is completed. The sender should wait for this greeting message before sending any commands.

Note: all the greeting type replies have the official name of the server host as the first word following the reply code.

For example,

```
220 <SP> USC-ISIF.ARPA <SP> Service ready <CRLF>
```

The table below lists alternative success and failure replies for each command. These must be strictly adhered to; a receiver may substitute text in the replies, but the meaning and action implied by the code numbers and by the specific command reply sequence cannot be altered.

## COMMAND-REPLY SEQUENCES

Each command is listed with its possible replies. The prefixes used before the possible replies are "P" for preliminary (not used in SMTP), "I" for intermediate, "S" for success, "F" for failure, and "E" for error. The 421 reply (service not available, closing transmission channel) may be given to any command if the SMTP-receiver knows it must shut down. This listing forms the basis for the State Diagrams in Section 4.4.

## CONNECTION ESTABLISHMENT

S: 220

F: 421

## HELO

S: 250

E: 500, 501, 504, 421

## MAIL

S: 250

F: 552, 451, 452

E: 500, 501, 421

RCPT  
S: 250, 251  
F: 550, 551, 552, 553, 450, 451, 452  
E: 500, 501, 503, 421

DATA  
I: 354 -> data -> S: 250  
F: 552, 554, 451, 452  
E: 500, 501, 503, 421

RSET  
S: 250  
E: 500, 501, 504, 421

SEND  
S: 250  
F: 552, 451, 452  
E: 500, 501, 502, 421

SOML  
S: 250  
F: 552, 451, 452  
E: 500, 501, 502, 421

SAML  
S: 250  
F: 552, 451, 452  
E: 500, 501, 502, 421

VERFY  
S: 250, 251  
F: 550, 551, 553  
E: 500, 501, 502, 504, 421

EXPN  
S: 250  
F: 550  
E: 500, 501, 502, 504, 421

HELP  
S: 211, 214  
E: 500, 501, 502, 504, 421

NOOP  
S: 250  
E: 500, 421

QUIT  
S: 221  
E: 500

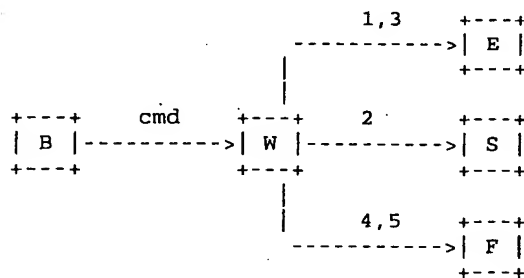
TURN  
S: 250  
F: 502  
E: 500, 503

## 4.4. STATE DIAGRAMS

Following are state diagrams for a simple-minded SMTP implementation. Only the first digit of the reply codes is used. There is one state diagram for each group of SMTP commands. The command groupings were determined by constructing a model for each command and then collecting together the commands with structurally identical models.

For each command there are three possible outcomes: "success" (S), "failure" (F), and "error" (E). In the state diagrams below we use the symbol B for "begin", and the symbol W for "wait for reply".

First, the diagram that represents most of the SMTP commands:

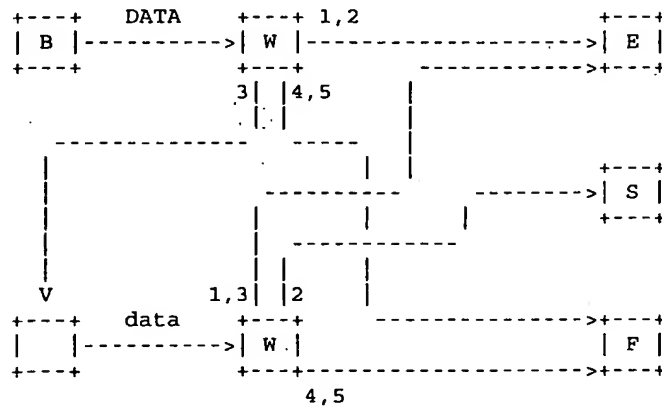


This diagram models the commands:

HELO, MAIL, RCPT, RSET, SEND, SOML, SAML, VRFY, EXPN, HELP,  
NOOP, QUIT, TURN.



A more complex diagram models the DATA command:



Note that the "data" here is a series of lines sent from the sender to the receiver with no response expected until the last line is sent.

## 4.5. DETAILS

## 4.5.1. MINIMUM IMPLEMENTATION

In order to make SMTP workable, the following minimum implementation is required for all receivers:

```
COMMANDS -- HELO
             MAIL
             RCPT
             DATA
             RSET
             NOOP
             QUIT
```

## 4.5.2. TRANSPARENCY

Without some provision for data transparency the character sequence "<CRLF>.<CRLF>" ends the mail text and cannot be sent by the user. In general, users are not aware of such "forbidden" sequences. To allow all user composed text to be transmitted transparently the following procedures are used.

1. Before sending a line of mail text the sender-SMTP checks the first character of the line. If it is a period, one additional period is inserted at the beginning of the line.
2. When a line of mail text is received by the receiver-SMTP it checks the line. If the line is composed of a single period it is the end of mail. If the first character is a period and there are other characters on the line, the first character is deleted.

The mail data may contain any of the 128 ASCII characters. All characters are to be delivered to the recipient's mailbox including format effectors and other control characters. If the transmission channel provides an 8-bit byte (octets) data stream, the 7-bit ASCII codes are transmitted right justified in the octets with the high order bits cleared to zero.

In some systems it may be necessary to transform the data as it is received and stored. This may be necessary for hosts that use a different character set than ASCII as their local character set, or that store data in records rather than

strings. If such transforms are necessary, they must be reversible -- especially if such transforms are applied to mail being relayed.

#### 4.5.3. SIZES

There are several objects that have required minimum maximum sizes. That is, every implementation must be able to receive objects of at least these sizes, but must not send objects larger than these sizes.

```
*****
*
* TO THE MAXIMUM EXTENT POSSIBLE, IMPLEMENTATION *
* TECHNIQUES WHICH IMPOSE NO LIMITS ON THE LENGTH *
* OF THESE OBJECTS SHOULD BE USED.                *
*
*****
```

##### user

The maximum total length of a user name is 64 characters.

##### domain

The maximum total length of a domain name or number is 64 characters.

##### path

The maximum total length of a reverse-path or forward-path is 256 characters (including the punctuation and element separators).

##### command line

The maximum total length of a command line including the command word and the <CRLF> is 512 characters.

##### reply line

The maximum total length of a reply line including the reply code and the <CRLF> is 512 characters.

**text line**

The maximum total length of a text line including the <CRLF> is 1000 characters (but not counting the leading dot duplicated for transparency).

**recipients buffer**

The maximum total number of recipients that must be buffered is 100 recipients.

```
*****
*
* TO THE MAXIMUM EXTENT POSSIBLE, IMPLEMENTATION *
* TECHNIQUES WHICH IMPOSE NO LIMITS ON THE LENGTH *
* OF THESE OBJECTS SHOULD BE USED. *
*
*****
```

Errors due to exceeding these limits may be reported by using the reply codes, for example:

500 Line too long.

501 Path too long

552 Too many recipients.

552 Too much mail data.

## APPENDIX A

### TCP Transport service

The Transmission Control Protocol [3] is used in the ARPA Internet, and in any network following the US DoD standards for internetwork protocols.

#### Connection Establishment

The SMTP transmission channel is a TCP connection established between the sender process port U and the receiver process port L. This single full duplex connection is used as the transmission channel. This protocol is assigned the service port 25 (31 octal), that is L=25.

#### Data Transfer

The TCP connection supports the transmission of 8-bit bytes. The SMTP data is 7-bit ASCII characters. Each character is transmitted as an 8-bit byte with the high-order bit cleared to zero.

APPENDIX B

NCP Transport service

The ARPANET Host-to-Host Protocol [4] (implemented by the Network Control Program) may be used in the ARPANET.

Connection Establishment

The SMTP transmission channel is established via NCP between the sender process socket U and receiver process socket L. The Initial Connection Protocol [5] is followed resulting in a pair of simplex connections. This pair of connections is used as the transmission channel. This protocol is assigned the contact socket 25 (31 octal), that is L=25.

Data Transfer

The NCP data connections are established in 8-bit byte mode. The SMTP data is 7-bit ASCII characters. Each character is transmitted as an 8-bit byte with the high-order bit cleared to zero.

APPENDIX C

NITS

The Network Independent Transport Service [6] may be used.

Connection Establishment

The SMTP transmission channel is established via NITS between the sender process and receiver process. The sender process executes the CONNECT primitive, and the waiting receiver process executes the ACCEPT primitive.

Data Transfer

The NITS connection supports the transmission of 8-bit bytes. The SMTP data is 7-bit ASCII characters. Each character is transmitted as an 8-bit byte with the high-order bit cleared to zero.

APPENDIX D

X.25 Transport service

It may be possible to use the X.25 service [7] as provided by the Public Data Networks directly, however, it is suggested that a reliable end-to-end protocol such as TCP be used on top of X.25 connections.



## APPENDIX E

### Theory of Reply Codes

The three digits of the reply each have a special significance. The first digit denotes whether the response is good, bad or incomplete. An unsophisticated sender-SMTP will be able to determine its next action (proceed as planned, redo, retrench, etc.) by simply examining this first digit. A sender-SMTP that wants to know approximately what kind of error occurred (e.g., mail system error, command syntax error) may examine the second digit, reserving the third digit for the finest gradation of information.

There are five values for the first digit of the reply code:

#### 1yz Positive Preliminary reply

The command has been accepted, but the requested action is being held in abeyance, pending confirmation of the information in this reply. The sender-SMTP should send another command specifying whether to continue or abort the action.

[Note: SMTP does not have any commands that allow this type of reply, and so does not have the continue or abort commands.]

#### 2yz Positive Completion reply

The requested action has been successfully completed. A new request may be initiated.

#### 3yz Positive Intermediate reply

The command has been accepted, but the requested action is being held in abeyance, pending receipt of further information. The sender-SMTP should send another command specifying this information. This reply is used in command sequence groups.

#### 4yz Transient Negative Completion reply

The command was not accepted and the requested action did not occur. However, the error condition is temporary and the action may be requested again. The sender should

return to the beginning of the command sequence (if any). It is difficult to assign a meaning to "transient" when two different sites (receiver- and sender- SMTPs) must agree on the interpretation. Each reply in this category might have a different time value, but the sender-SMTP is encouraged to try again. A rule of thumb to determine if a reply fits into the 4yz or the 5yz category (see below) is that replies are 4yz if they can be repeated without any change in command form or in properties of the sender or receiver. (E.g., the command is repeated identically and the receiver does not put up a new implementation.)

5yz Permanent Negative Completion reply

The command was not accepted and the requested action did not occur. The sender-SMTP is discouraged from repeating the exact request (in the same sequence). Even some "permanent" error conditions can be corrected, so the human user may want to direct the sender-SMTP to reinitiate the command sequence by direct action at some point in the future (e.g., after the spelling has been changed, or the user has altered the account status).

The second digit encodes responses in specific categories:

- x0z Syntax -- These replies refer to syntax errors, syntactically correct commands that don't fit any functional category, and unimplemented or superfluous commands.
- x1z Information -- These are replies to requests for information, such as status or help.
- x2z Connections -- These are replies referring to the transmission channel.
- x3z Unspecified as yet.
- x4z Unspecified as yet.
- x5z Mail system -- These replies indicate the status of the receiver mail system vis-a-vis the requested transfer or other mail system action.

The third digit gives a finer gradation of meaning in each category specified by the second digit. The list of replies

illustrates this. Each reply text is recommended rather than mandatory, and may even change according to the command with which it is associated. On the other hand, the reply codes must strictly follow the specifications in this section. Receiver implementations should not invent new codes for slightly different situations from the ones described here, but rather adapt codes already defined.

For example, a command such as NOOP whose successful execution does not offer the sender-SMTP any new information will return a 250 reply. The response is 502 when the command requests an unimplemented non-site-specific action. A refinement of that is the 504 reply for a command that is implemented, but that requests an unimplemented parameter.

The reply text may be longer than a single line; in these cases the complete text must be marked so the sender-SMTP knows when it can stop reading the reply. This requires a special format to indicate a multiple line reply.

The format for multiline replies requires that every line, except the last, begin with the reply code, followed immediately by a hyphen, "-" (also known as minus), followed by text. The last line will begin with the reply code, followed immediately by <SP>, optionally some text, and <CRLF>.

For example:

```
123-First line
123-Second line
123-234 text beginning with numbers
123 The last line
```

In many cases the sender-SMTP then simply needs to search for the reply code followed by <SP> at the beginning of a line, and ignore all preceding lines. In a few cases, there is important data for the sender in the reply "text". The sender will know these cases from the current context.

## APPENDIX F

## Scenarios

This section presents complete scenarios of several types of SMTP sessions.

## A Typical SMTP Transaction Scenario

This SMTP example shows mail sent by Smith at host USC-ISIF, to Jones, Green, and Brown at host BBN-UNIX. Here we assume that host USC-ISIF contacts host BBN-UNIX directly. The mail is accepted for Jones and Brown. Green does not have a mailbox at host BBN-UNIX.

-----

```
R: 220 BBN-UNIX.ARPA Simple Mail Transfer Service Ready
S: HELO USC-ISIF.ARPA
R: 250 BBN-UNIX.ARPA

S: MAIL FROM:<Smith@USC-ISIF.ARPA>
R: 250 OK

S: RCPT TO:<Jones@BBN-UNIX.ARPA>
R: 250 OK

S: RCPT TO:<Green@BBN-UNIX.ARPA>
R: 550 No such user here

S: RCPT TO:<Brown@BBN-UNIX.ARPA>
R: 250 OK

S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: .
R: 250 OK

S: QUIT
R: 221 BBN-UNIX.ARPA Service closing transmission channel
```

Scenario 1

-----

Aborted SMTP Transaction Scenario

```
-----  
R: 220 MIT-Multics.ARPA Simple Mail Transfer Service Ready  
S: HELO ISI-VAXA.ARPA  
R: 250 MIT-Multics.ARPA  
  
S: MAIL FROM:<Smith@ISI-VAXA.ARPA>  
R: 250 OK  
  
S: RCPT TO:<Jones@MIT-Multics.ARPA>  
R: 250 OK  
  
S: RCPT TO:<Green@MIT-Multics.ARPA>  
R: 550 No such user here  
  
S: RSET  
R: 250 OK  
  
S: QUIT  
R: 221 MIT-Multics.ARPA Service closing transmission channel
```

Scenario 2

```
-----
```

## Relayed Mail Scenario

-----  
Step 1 -- Source Host to Relay Host

R: 220 USC-ISIE.ARPA Simple Mail Transfer Service Ready  
S: HELO MIT-AI.ARPA  
R: 250 USC-ISIE.ARPA  
  
S: MAIL FROM:<JQP@MIT-AI.ARPA>  
R: 250 OK  
  
S: RCPT TO:<@USC-ISIE.ARPA:Jones@BBN-VAX.ARPA>  
R: 250 OK  
  
S: DATA  
R: 354 Start mail input; end with <CRLF>.<CRLF>  
S: Date: 2 Nov 81 22:33:44  
S: From: John Q. Public <JQP@MIT-AI.ARPA>  
S: Subject: The Next Meeting of the Board  
S: To: Jones@BBN-Vax.ARPA  
S:  
S: Bill:  
S: The next meeting of the board of directors will be  
S: on Tuesday.  
S: John.  
S: .  
R: 250 OK  
  
S: QUIT  
R: 221 USC-ISIE.ARPA Service closing transmission channel

Step 2 -- Relay Host to Destination Host

R: 220 BBN-VAX.ARPA Simple Mail Transfer Service Ready  
S: HELO USC-ISIE.ARPA  
R: 250 BBN-VAX.ARPA

S: MAIL FROM:<@USC-ISIE.ARPA:JQP@MIT-AI.ARPA>  
R: 250 OK

S: RCPT TO:<Jones@BBN-VAX.ARPA>  
R: 250 OK

S: DATA  
R: 354 Start mail input; end with <CRLF>.<CRLF>  
S: Received: from MIT-AI.ARPA by USC-ISIE.ARPA ;  
2 Nov 81 22:40:10 UT  
S: Date: 2 Nov 81 22:33:44  
S: From: John Q. Public <JQP@MIT-AI.ARPA>  
S: Subject: The Next Meeting of the Board  
S: To: Jones@BBN-Vax.ARPA

S:  
S: Bill:  
S: The next meeting of the board of directors will be  
S: on Tuesday.  
S: John.  
S: .  
R: 250 OK

S: QUIT  
R: 221 USC-ISIE.ARPA Service closing transmission channel

Scenario 3

-----

## Verifying and Sending Scenario

-----

R: 220 SU-SCORE.ARPA Simple Mail Transfer Service Ready  
S: HELO MIT-MC.ARPA  
R: 250 SU-SCORE.ARPA

S: VRFY Crispin  
R: 250 Mark Crispin <Admin.MRC@SU-SCORE.ARPA>

S: SEND FROM:<EAK@MIT-MC.ARPA>  
R: 250 OK

S: RCPT TO:<Admin.MRC@SU-SCORE.ARPA>  
R: 250 OK

S: DATA  
R: 354 Start mail input; end with <CRLF>.<CRLF>  
S: Blah blah blah...  
S: ...etc. etc. etc.  
S: .  
R: 250 OK

S: QUIT  
R: 221 SU-SCORE.ARPA Service closing transmission channel

## Scenario 4



### Sending and Mailing Scenarios

First the user's name is verified, then an attempt is made to send to the user's terminal. When that fails, the message is mailed to the user's mailbox.

```
-----  
R: 220 SU-SCORE.ARPA Simple Mail Transfer Service Ready  
S: HELO MIT-MC.ARPA  
R: 250 SU-SCORE.ARPA  
  
S: VRFY Crispin  
R: 250 Mark Crispin <Admin.MRC@SU-SCORE.ARPA>  
  
S: SEND FROM:<EAK@MIT-MC.ARPA>  
R: 250 OK  
  
S: RCPT TO:<Admin.MRC@SU-SCORE.ARPA>  
R: 450 User not active now  
  
S: RSET  
R: 250 OK  
  
S: MAIL FROM:<EAK@MIT-MC.ARPA>  
R: 250 OK  
  
S: RCPT TO:<Admin.MRC@SU-SCORE.ARPA>  
R: 250 OK  
  
S: DATA  
R: 354 Start mail input; end with <CRLF>.<CRLF>  
S: Blah blah blah...  
S: ...etc. etc. etc.  
S: .  
R: 250 OK  
  
S: QUIT  
R: 221 SU-SCORE.ARPA Service closing transmission channel
```

### Scenario 5

-----

Doing the preceding scenario more efficiently.

-----

R: 220 SU-SCORE.ARPA Simple Mail Transfer Service Ready  
S: HELO MIT-MC.ARPA  
R: 250 SU-SCORE.ARPA  
S: VRFY Crispin  
R: 250 Mark Crispin <Admin.MRC@SU-SCORE.ARPA>  
S: SOML FROM:<EAK@MIT-MC.ARPA>  
R: 250 OK  
S: RCPT TO:<Admin.MRC@SU-SCORE.ARPA>  
R: 250 User not active now, so will do mail.  
S: DATA  
R: 354 Start mail input; end with <CRLF>.<CRLF>  
S: Blah blah blah...  
S: ...etc. etc. etc.  
S: .  
R: 250 OK  
S: QUIT  
R: 221 SU-SCORE.ARPA Service closing transmission channel

Scenario 6

-----

### Mailing List Scenario

First each of two mailing lists are expanded in separate sessions with different hosts. Then the message is sent to everyone that appeared on either list (but no duplicates) via a relay host.

-----  
Step 1 -- Expanding the First List

```
R: 220 MIT-AI.ARPA Simple Mail Transfer Service Ready
S: HELO SU-SCORE.ARPA
R: 250 MIT-AI.ARPA

S: EXPN Example-People
R: 250-<ABC@MIT-MC.ARPA>
R: 250-Fred Fonebone <Fonebone@USC-ISIQ.ARPA>
R: 250-Xenon Y. Zither <XYZ@MIT-AI.ARPA>
R: 250-Quincy Smith <@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>
R: 250-<joe@foo-unix.ARPA>
R: 250 <xyz@bar-unix.ARPA>

S: QUIT
R: 221 MIT-AI.ARPA Service closing transmission channel
```

Step 2 -- Expanding the Second List

R: 220 MIT-MC.ARPA Simple Mail Transfer Service Ready  
S: HELO SU-SCORE.ARPA  
R: 250 MIT-MC.ARPA  
  
S: EXPN Interested-Parties  
R: 250-Al Calico <ABC@MIT-MC.ARPA>  
R: 250-<XYZ@MIT-AI.ARPA>  
R: 250-Quincy Smith <@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>  
R: 250-<fred@BBN-UNIX.ARPA>  
R: 250 <xyz@bar-unix.ARPA>  
  
S: QUIT  
R: 221 MIT-MC.ARPA Service closing transmission channel

Step 3 -- Mailing to All via a Relay Host

```
R: 220 USC-ISIE.ARPA Simple Mail Transfer Service Ready
S: HELO SU-SCORE.ARPA
R: 250 USC-ISIE.ARPA

S: MAIL FROM:<Account.Person@SU-SCORE.ARPA>
R: 250 OK
S: RCPT TO:<@USC-ISIE.ARPA:ABC@MIT-MC.ARPA>
R: 250 OK
S: RCPT TO:<@USC-ISIE.ARPA:Fonebone@USC-ISIQA.ARPA>
R: 250 OK
S: RCPT TO:<@USC-ISIE.ARPA:XYZ@MIT-AI.ARPA>
R: 250 OK
S: RCPT
  TO:<@USC-ISIE.ARPA,@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>
R: 250 OK
S: RCPT TO:<@USC-ISIE.ARPA:joe@FOO-UNIX.ARPA>
R: 250 OK
S: RCPT TO:<@USC-ISIE.ARPA:xyz@BAR-UNIX.ARPA>
R: 250 OK
S: RCPT TO:<@USC-ISIE.ARPA:fred@BBN-UNIX.ARPA>
R: 250 OK

S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: .
R: 250 OK

S: QUIT
R: 221 USC-ISIE.ARPA Service closing transmission channel
```

Scenario 7

-----

## Forwarding Scenarios

-----

R: 220 USC-ISIF.ARPA Simple Mail Transfer Service Ready  
S: HELO LBL-UNIX.ARPA  
R: 250 USC-ISIF.ARPA

S: MAIL FROM:<mo@LBL-UNIX.ARPA>  
R: 250 OK

S: RCPT TO:<fred@USC-ISIF.ARPA>  
R: 251 User not local; will forward to <Jones@USC-ISI.ARPA>

S: DATA  
R: 354 Start mail input; end with <CRLF>.<CRLF>  
S: Blah blah blah...  
S: ...etc. etc. etc.  
S: .  
R: 250 OK

S: QUIT  
R: 221 USC-ISIF.ARPA Service closing transmission channel

## Scenario 8

---

Step 1 -- Trying the Mailbox at the First Host

R: 220 USC-ISIF.ARPA Simple Mail Transfer Service Ready  
S: HELO LBL-UNIX.ARPA  
R: 250 USC-ISIF.ARPA  
  
S: MAIL FROM:<mo@LBL-UNIX.ARPA>  
R: 250 OK  
  
S: RCPT TO:<fred@USC-ISIF.ARPA>  
R: 251 User not local; will forward to <Jones@USC-ISI.ARPA>  
  
S: RSET  
R: 250 OK  
  
S: QUIT  
R: 221 USC-ISIF.ARPA Service closing transmission channel

Step 2 -- Delivering the Mail at the Second Host

R: 220 USC-ISI.ARPA Simple Mail Transfer Service Ready  
S: HELO LBL-UNIX.ARPA  
R: 250 USC-ISI.ARPA  
  
S: MAIL FROM:<mo@LBL-UNIX.ARPA>  
R: 250 OK  
  
S: RCPT TO:<Jones@USC-ISI.ARPA>  
R: OK  
  
S: DATA  
R: 354 Start mail input; end with <CRLF>.<CRLF>  
S: Blah blah blah...  
S: ...etc. etc. etc.  
S: .  
R: 250 OK  
  
S: QUIT  
R: 221 USC-ISI.ARPA Service closing transmission channel

Scenario 9

---

## Too Many Recipients Scenario

-----

R: 220 BERKELEY.ARPA Simple Mail Transfer Service Ready  
S: HELO USC-ISIF.ARPA  
R: 250 BERKELEY.ARPA

S: MAIL FROM:<Postel@USC-ISIF.ARPA>  
R: 250 OK

S: RCPT TO:<fabry@BERKELEY.ARPA>  
R: 250 OK

S: RCPT TO:<eric@BERKELEY.ARPA>  
R: 552 Recipient storage full, try again in another transaction

S: DATA  
R: 354 Start mail input; end with <CRLF>.<CRLF>  
S: Blah blah blah...  
S: ...etc. etc. etc.  
S: .  
R: 250 OK

S: MAIL FROM:<Postel@USC-ISIF.ARPA>  
R: 250 OK

S: RCPT TO:<eric@BERKELEY.ARPA>  
R: 250 OK

S: DATA  
R: 354 Start mail input; end with <CRLF>.<CRLF>  
S: Blah blah blah...  
S: ...etc. etc. etc.  
S: .  
R: 250 OK

S: QUIT  
R: 221 BERKELEY.ARPA Service closing transmission channel

## Scenario 10

-----

Note that a real implementation must handle many recipients as specified in Section 4.5.3.



GLOSSARY

ASCII

American Standard Code for Information Interchange [1].

command

A request for a mail service action sent by the sender-SMTP to the receiver-SMTP.

domain

The hierarchially structured global character string address of a host computer in the mail system.

end of mail data indication

A special sequence of characters that indicates the end of the mail data. In particular, the five characters carriage return, line feed, period, carriage return, line feed, in that order.

host

A computer in the internetwork environment on which mailboxes or SMTP processes reside.

line

A a sequence of ASCII characters ending with a <CRLF>.

mail data

A sequence of ASCII characters of arbitrary length, which conforms to the standard set in the Standard for the Format of ARPA Internet Text Messages (RFC 822 [2]).

mailbox

A character string (address) which identifies a user to whom mail is to be sent. Mailbox normally consists of the host and user specifications. The standard mailbox naming convention is defined to be "user@domain". Additionally, the "container" in which mail is stored.

**receiver-SMTP process**

A process which transfers mail in cooperation with a sender-SMTP process. It waits for a connection to be established via the transport service. It receives SMTP commands from the sender-SMTP, sends replies, and performs the specified operations.

**reply**

A reply is an acknowledgment (positive or negative) sent from receiver to sender via the transmission channel in response to a command. The general form of a reply is a completion code (including error codes) followed by a text string. The codes are for use by programs and the text is usually intended for human users.

**sender-SMTP process**

A process which transfers mail in cooperation with a receiver-SMTP process. A local language may be used in the user interface command/reply dialogue. The sender-SMTP initiates the transport service connection. It initiates SMTP commands, receives replies, and governs the transfer of mail.

**session**

The set of exchanges that occur while the transmission channel is open.

**transaction**

The set of exchanges required for one message to be transmitted for one or more recipients.

**transmission channel**

A full-duplex communication path between a sender-SMTP and a receiver-SMTP for the exchange of commands, replies, and mail text.

**transport service**

Any reliable stream-oriented data communication services. For example, NCP, TCP, NITS.

user

A human being (or a process on behalf of a human being) wishing to obtain mail transfer service. In addition, a recipient of computer mail.

word

A sequence of printing characters.

<CRLF>

The characters carriage return and line feed (in that order).

<SP>

The space character.

## REFERENCES

## [1] ASCII

ASCII, "USA Code for Information Interchange", United States of America Standards Institute, X3.4, 1968. Also in: Feinler, E. and J. Postel, eds., "ARPANET Protocol Handbook", NIC 7104, for the Defense Communications Agency by SRI International, Menlo Park, California, Revised January 1978.

## [2] RFC 822

Crocker, D., "Standard for the Format of ARPA Internet Text Messages," RFC 822, Department of Electrical Engineering, University of Delaware, August 1982.

## [3] TCP

Postel, J., ed., "Transmission Control Protocol - DARPA Internet Program Protocol Specification", RFC 793, USC/Information Sciences Institute, NTIS AD Number A111091, September 1981. Also in: Feinler, E. and J. Postel, eds., "Internet Protocol Transition Workbook", SRI International, Menlo Park, California, March 1982.

## [4] NCP

McKenzie, A., "Host/Host Protocol for the ARPA Network", NIC 8246, January 1972. Also in: Feinler, E. and J. Postel, eds., "ARPANET Protocol Handbook", NIC 7104, for the Defense Communications Agency by SRI International, Menlo Park, California, Revised January 1978.

## [5] Initial Connection Protocol

Postel, J., "Official Initial Connection Protocol", NIC 7101, 11 June 1971. Also in: Feinler, E. and J. Postel, eds., "ARPANET Protocol Handbook", NIC 7104, for the Defense Communications Agency by SRI International, Menlo Park, California, Revised January 1978.

## [6] NITS

PSS/SG3, "A Network Independent Transport Service", Study Group 3, The Post Office PSS Users Group, February 1980. Available from the DCPU, National Physical Laboratory, Teddington, UK.

[7] X.25

CCITT, "Recommendation X.25 - Interface Between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks," CCITT Orange Book, Vol. VIII.2, International Telephone and Telegraph Consultative Committee, Geneva, 1976.

# 甲 7 号 証

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-242326

(43)公開日 平成8年(1996)9月17日

(51)Int.Cl. <sup>9</sup>	識別記号	庁内整理番号	FI	技術表示箇所
H04N 1/00	107		H04N 1/00	107Z

審査請求 未請求 請求項の数20 OL (全 15 頁)

(21)出願番号 特願平7-45847

(22)出願日 平成7年(1995)3月6日

(71)出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72)発明者 豊田 清

大阪府門真市大字門真1006番地 松下電器  
産業株式会社内

(72)発明者 坂東 達夫

大阪府門真市大字門真1006番地 松下電器  
産業株式会社内

(72)発明者 沢田 利久

大阪府門真市大字門真1006番地 松下電器  
産業株式会社内

(74)代理人 弁理士 藤合 正博

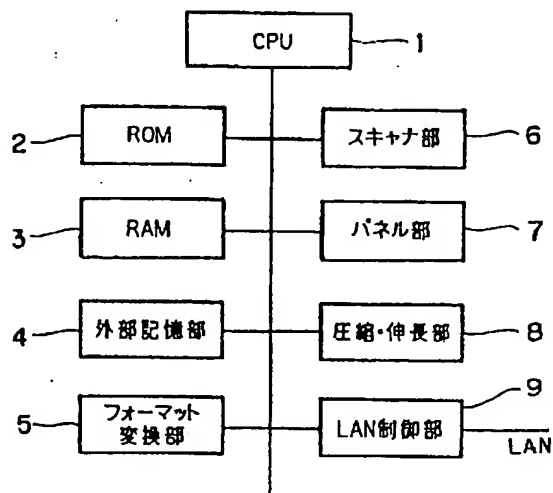
(54)【発明の名称】 ファクシミリ型電子メール装置

(57)【要約】

ファクシミリ型電子メール装置

【目的】 ファクシミリと同様の簡単な操作でイメージデータを電子メールとして送信する。

【構成】 CPU1は本装置の制御を行う。ROM2にはプログラムが記憶され、RAM3はプログラムのデータ用として使用される。外部記憶部4は圧縮されたイメージデータが蓄積され、フォーマット変換部5は圧縮されたイメージデータを電子メールのフォーマットに変換する。スキャナ部6はイメージを読み込み、パネル部7はイメージ読み込みの指示、電子メールの宛先入力を行う。圧縮・伸長部8は読み込まれたイメージデータを圧縮する。LAN制御部9はLAN(ローカルエリアネットワーク)と接続され、インターネットの電子メールデータの送受信を行う。



## 【特許請求の範囲】

【請求項1】 原稿を読み取ってイメージデータに変換する手段と、前記読み取ったイメージデータを圧縮する手段と、前記圧縮されたイメージデータを電子メールのフォーマットに変換する手段と、電子メールの宛先アドレスを入力する手段と、前記入力された宛先アドレスに対して前記フォーマット変換されたイメージデータを電子メールが可能なネットワークに送信する手段とを備えたファクシミリ型電子メール装置。

【請求項2】 電子メールにより送信されてきたイメージデータを受信する手段と、前記イメージデータを電子メールのフォーマットからファクシミリのフォーマットに変換する手段と、前記フォーマット変換されたイメージデータを伸長する手段と、前記伸長されたイメージデータを印刷する手段とを備えた請求項1に記載のファクシミリ型電子メール装置。

【請求項3】 文字コードデータを電子メールにより送受信する手段とを備えた請求項1または2に記載のファクシミリ型電子メール装置。

【請求項4】 電子メールの宛先アドレスの入力手段として、イメージデータの所定の位置に記入された電子メールの宛先アドレスを認識する手段を備えた請求項1から3のいずれかに記載のファクシミリ型電子メール装置。

【請求項5】 送信者の任意の電子メールアドレスを電子メールの発信元アドレスとして入力し、送信が失敗したときに前記発信元アドレスにその旨を通知する手段を備えた請求項1から4のいずれかに記載のファクシミリ型電子メール装置。

【請求項6】 音声を入出力する手段と、音声データをイメージデータと対応づける手段と、前記対応づけされた音声データとイメージデータを一つの電子メールとして送受信する手段とを備えた請求項1から5のいずれかに記載のファクシミリ型電子メール装置。

【請求項7】 動画を入出力する手段と、映像データとイメージデータを対応づける手段と、前記対応づけされた映像データとイメージデータを一つの電子メールとして送受信する手段とを備えた請求項1から6のいずれかに記載のファクシミリ型電子メール装置。

【請求項8】 受信した文字コードデータの所定の位置にある文字列を解析する手段と、前記解析結果に応じた処理を行う手段とを備えた請求項3に記載のファクシミリ型電子メール装置。

【請求項9】 解析した文字列が宛先メールアドレスの登録を意味する文字列である場合、続く文字列の中の送信者IDと複数の宛先メールアドレスとを対応させて登録するとともに、メール送信時に送信者IDを入力したときに、前記入力された送信者IDと前記登録された送信者IDとが一致した場合には、前記対応した宛先メールアドレスを表示し、その中から必要な宛先を選択する

ことを特徴とする請求項8に記載のファクシミリ型電子メール装置。

【請求項10】 解析した文字列が発信元メールアドレスの登録を意味する文字列である場合、続く文字列の中の送信者IDと発信元メールアドレスとを対応させて登録するとともに、メール送信時に送信者IDを入力したときに、前記入力された送信者IDと前記登録された送信者IDとが一致した場合には、前記対応した発信元メールアドレスを発信元として入力することを特徴とする請求項8に記載のファクシミリ型電子メール装置。

【請求項11】 メール送信時に送信者IDを入力したときに、前記入力された送信者IDと登録された送信者IDとが一致した場合には、前記送信者IDに対応した発信元メールアドレスを発信元としてメールのヘッダに書き込む手段を備えた請求項9および10に記載のファクシミリ型電子メール装置。

【請求項12】 電話回線を使ってファクシミリの手順でイメージデータを送受信する手段と、イメージデータを電話回線を使ってファクシミリ手順で送信するか、ネットワークを使って電子メールの手順で送信するかを選択する手段とを備えた請求項2から11のいずれかに記載のファクシミリ型電子メール装置。

【請求項13】 電子メールとして受信した文字コードデータの所定の位置にある文字列を解析した結果、解析した文字列がファクシミリデータ受信待ちを意味する文字列である場合に、続く文字列の中の通知用電子メールアドレスを認識し、前記通知用電子メールアドレスを認識した後に電話回線からファクシミリ手順でイメージデータを受信したときには、前記通知用電子メールアドレスに受信通知の電子メールを送信することを特徴とする請求項12に記載のファクシミリ型電子メール装置。

【請求項14】 電子メールとして受信したイメージデータのヘッダの所定の位置の文字列を解析した結果、解析した文字列がファクシミリ送信を意味する文字列である場合に、前記文字列以外のヘッダの所定の位置の文字列をファクシミリ番号と認識し、前記受信したイメージデータを電話回線を通してファクシミリ手順で前記ファクシミリ番号に送信することを特徴とする請求項12または13に記載のファクシミリ型電子メール装置。

【請求項15】 送信する時刻に対応する世界各国の現地時間を計算する手段を備え、宛先アドレスから送信する国を識別するとともに、前記識別した国に対応する現地時刻にイメージデータを電子メールとして送信することを特徴とする請求項1から14のいずれかに記載のファクシミリ型電子メール装置。

【請求項16】 自装置宛の電子メールアドレスを複数登録する手段と、電子メールにより送信されてきたイメージデータを前記電子メールアドレスごとに蓄積、管理する手段と、各電子メールアドレス毎にパスワードを設定する手段と、前記パスワードを入力したときに前記蓄

積されたイメージデータを出力する手段とを備えた請求項2から15のいずれかに記載のファクシミリ型電子メール装置。

【請求項17】 自装置の各メールアドレス毎に少なくとも各電子メールアドレスを記載したカバーシートを付加して出力する手段を備えた請求項16に記載のファクシミリ型電子メール装置。

【請求項18】 電子メールを受信したときに、送信者の電子メールアドレスに前記電子メールが届いた旨の確認電子メールを送信する手段を備えた請求項2から17のいずれかに記載のファクシミリ型電子メール装置。

【請求項19】 受信電子メール毎に受付IDを記載して出力する手段と、前記受信電子メールの送信元電子メールアドレスと受付IDを対応させる手段と、前記受付IDを入力したときに前記送信元電子メールアドレスに確認電子メールを送信する手段を備えた請求項2から18のいずれかに記載のファクシミリ型電子メール装置。

【請求項20】 受信電子メールの送信が失敗したときに、その旨を印刷する手段と、イメージデータの1ページ目を出力する手段を備えた請求項2から19のいずれかに記載のファクシミリ型電子メール装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、イメージデータをファクシミリと同様の簡単な操作で送受信できるファクシミリ型電子メール装置に関するものである。

【0002】

【従来の技術】 近年、ファクシミリは、イメージを簡単な操作で送受信できる装置として、ほとんどすべての企業に普及してきている。しかし、送信したイメージが目的とする人に届いたかどうか分からないこと、受信した原稿は誰でも読めてしまうこと、回線状況の悪い外国に送信するときは、失敗する度に送り直す必要があることなどの問題は解決されておらず、ファクシミリを使用する人もあきらめている傾向がある。

【0003】 一方、世界規模のネットワークであるインターネットが現在急速に増殖中であり、インターネットを使った電子メールを使う人の数も企業を中心に急激に伸びている。電子メールは、個人に属するパソコン（以下PCと呼ぶ）やワークステーション（以下WSと呼ぶ）を使って送受信するものであり、前に述べたファクシミリの問題点はない。また、エラーに関しては、信頼性のあるプロトコルTCP/IPによって、データエラーは基本的にはなくなる。

【0004】 しかし、電子メールで送受信するデータは、キーボードから入力された文字コードが主体であり、イメージデータはまだ一般的ではない。その理由は、イメージデータを入力して電子メールとして送信できるフォーマットに変換するのに手間がかかるためと考えられる。つまり、イメージデータを電子メールとして

送信するための操作性が悪いのである。

【0005】 そこで、操作性を改善するために、電子メールを利用したファクシミリ装置が例えば特開平2-172348号公報等に提案されている。

【0006】 図28は従来の電子メールを利用したファクシミリ装置の概略構成図を示すものである。101はCPUであり、装置の制御を行う。102はROMであり、プログラムの記憶される。103はRAMであり、プログラムのデータ用として使用される。104は電話回線に接続されたモデム部である。105はプリンタ部であり、イメージデータを印刷する。106はスキャナ部であり、イメージを読み込む。107はパネル部であり、イメージ読み込みの指示、送信者の識別情報入力、受信者の識別情報入力を行う。108は圧縮・伸長部であり、イメージデータの圧縮または伸長を行う。109はコンピュータI/F部であり、ホストコンピュータと接続され、電子メールの送受信を行う。

【0007】 以下、ファクシミリ送信するときの動作について説明する。スキャナ部106に原稿を置き、パネル部107から宛先の電話番号、送信者の識別コードを入力し、スタートボタンを押す。スキャナ部106は、イメージを読み込み、圧縮・伸長部108でイメージデータを圧縮し、モデム部104を通じて電話回線から宛先にイメージデータがファクシミリ手順で送信される。送信結果は、コンピュータI/F部109からホストコンピュータを介して送信者の端末に文字コードの電子メールとして送信される。

【0008】 次に、ファクシミリ原稿を受信するときの動作について説明する。まず、受信者は利用者識別情報をパネル部107からあらかじめ入力し、RAM103に記憶しておく。ファクシミリ原稿をモデム部104を介して受信したときに、ファクシミリ手順の中で、宛先情報が送られてきたときは、RAM103に記憶してある利用者識別情報と一致しているかどうかを確認し、一致していれば、その利用者の端末にファクシミリ原稿を受信したことをコンピュータI/F部109を通じて電子メールで知らせる。受信したファクシミリのイメージデータは、プリンタ部105で印刷される。

【0009】

【発明が解決しようとする課題】 しかしながら、上記従来の構成では、相手端末にファクシミリ原稿の受信があったことは通知されるものの、送信したファクシミリデータが実際に目的とする本人に届いたかわからないこと、また送信側と受信側の両方に上記構成のファクシミリがなければ、ファクシミリ手順の中で宛先は確認できず、ファクシミリデータを受信したことが目的の受信者に通知されないこと、ファクシミリデータ自体はファクシミリ手順を使って通信するので、回線事情の悪い海外との通信の信頼性がない場合があることなどの問題を有していた。



【0010】本発明は、上記従来の問題を解決するもので、イメージデータを個人宛に送受信できる、操作性の優れたファクシミリ型電子メール装置を提供することを目的とする。

【0011】

【課題を解決するための手段】本発明は、上記目的を達成するために、原稿を読み取ったイメージデータを電子メールのフォーマットに変換してからネットワーク上に送信するようにしたものである。受信する際は、逆に電子メールのフォーマットをファクシミリのフォーマットに変換した後に、プリンタで印刷するようにしたものである。

【0012】本発明はまた、ネットワークを通じた電子メールによるイメージデータの送受信と、電話回線を通じたファクシミリによるイメージデータの送受信とを、いずれも可能にしたものである。

【0013】

【作用】したがって、本発明の電子メールによれば、本装置を電子メールが可能なネットワークに接続することにより、従来のファクシミリ装置のように、単に原稿を読み取らせるという簡単な操作だけで、イメージデータを相手側に送信することができ、受信する側は、単に装置を動作状態に置くだけで、イメージデータを受信することができる。しかも、従来のファクシミリ装置のように送信相手に確実に届いたかどうか分からなかったり、他人に読まれたり、通信エラーが発生するというような問題を解決できる利点がある。

【0014】本発明はまた、電子メールとファクシミリの両方で送受信できるようにすることにより、送信相手が電子メール装置である場合は電子メールにより、また送信相手がファクシミリ装置の場合はファクシミリ装置により送信することができる。さらに、受信した電子メールをファクシミリで送信したり、受信したファクシミリを電子メールで送信したりすることができる。

【0015】

【実施例】

(実施例1) 以下、本発明の第1の実施例について、図面を参照しながら説明する。図1はイメージデータをファクシミリと同様の簡単な操作で送信可能な電子メール装置の概略ブロック図である。図1において、1はCPUであり、本装置の制御を行う。2はROMであり、プログラムが記憶される。3はRAMであり、プログラムのデータ用として使用される。4はハードディスク等の外部記憶部であり、圧縮されたイメージデータが蓄積される。5はフォーマット変換部であり、圧縮されたイメージデータを電子メールのフォーマットに変換する。6はスキャナ部であり、イメージを読み込む。7はパネル部であり、イメージの読み込み指示、電子メールの宛先入力を行う。8は圧縮・伸長部であり、読み込まれたイメージデータの圧縮または送信されてきたイメージデー

タの伸長を行う。9はLAN制御部であり、LAN（ローカルエリアネットワーク）と接続され、インターネットの電子メールデータの送受信を行う。

【0016】図2は本実施例におけるイメージデータを送信する際のフローを示す。まずステップS1で、原稿をスキャナ部6にセットし、パネル部7から電子メールの宛先（英数字）を入力し、スタートボタンを押す。次にステップS2で、スキャナ部6にセットした原稿がイメージデータとして入力され、ステップS3で、圧縮・伸長部8によって圧縮され、外部記憶部4に蓄積される。このとき、原稿が複数ページあれば、複数ページを一つのファイルとして外部記憶部4に蓄積される。次にステップS4で、蓄積されたイメージデータがフォーマット変換部5によって、電子メールのフォーマットに変換され、外部記憶部4に蓄積される。そしてステップS5で、フォーマット変換されたイメージデータがLAN制御部9によって電子メールとして宛先に送信される。

【0017】図3にイメージデータの電子メールフォーマットへの変換方法を示す。スキャナ部6から読み込まれ、圧縮・伸長部8で圧縮されたイメージデータは、バイナリイメージデータ21であり、このままではインターネットの電子メールとして送信することはできない。インターネットの電子メールは、7bitの文字コードしか伝送手順上許していない。そこで、このバイナリイメージデータを7bitの文字コードに変換してテキストコード化イメージデータ22を作成する。その後、このデータに少なくとも宛先、発信元、データの形式、文字コードへの変換方式の入ったヘッダ23を付加する。

【0018】文字コードへの変換方式およびヘッダのフォーマットは、MIME(Multipurpose Internet Mail Extensions)と呼ばれるインターネットの電子メールの標準に従う。したがって、MIMEをサポートした電子メールのプログラムをインストールしている端末であれば、本装置からイメージデータを電子メールとして送信することができる。

【0019】このように、上記第1の実施例によれば、ファクシミリと同様の簡単な操作でイメージデータを電子メールとして送信できる。したがって、イメージデータを個人宛に送信でき、送信したデータを目的とする人以外は見ることができず、海外とのデータ通信の信頼性を向上させることができる。

【0020】(実施例2) 次に、本発明の第2の実施例について図4を参照して説明する。本実施例は、図1に示した第1の実施例の構成に、電子メールのイメージデータを印刷できる形に変換するフォーマット逆変換部10と、フォーマット逆変換されたイメージデータを印刷するためのプリンタ部11を追加したものである。

【0021】図5は本実施例におけるイメージデータを受信する際のフローを示す。まずステップS11で、LAN制御部9から電子メールのイメージデータを受信

し、外部記憶部4に一旦蓄積する。次にステップS12で、電子メールのイメージデータをフォーマット逆変換部10でファクシミリのフォーマットに変換し、ステップS13で、圧縮・伸長部8によって伸長して外部記憶部4に蓄積する。そしてステップS14で、蓄積されたイメージデータをプリンタ部11で印刷する。

【0022】逆フォーマット変換は、図3のイメージデータの電子メールフォーマットへの変換方法とは逆の手順をたどる。はじめに、ヘッダ23からこの電子メールがイメージデータであることを確認する。イメージデータでなければ、発信元に対してエラー通知の電子メールを送る。次にテキストコード化イメージデータ22をバイナリイメージデータ21に変換する。

【0023】このように、上記第2の実施例によれば、電子メールの個人アドレスを持っていない人でも、イメージデータを受信することができる。

【0024】(実施例3) 次に、本発明の第3の実施例について図6を参照して説明する。本実施例は、図4に示した第2の実施例の構成に、文字コードデータをイメージデータに変換するためのフォント部12を追加したものである。

【0025】図7は本実施例における文字コードデータを印刷する際のフローを示す。まずステップS21で、LAN制御部9により、受信した電子メールが文字コードデータかイメージデータかを、データのヘッダ部のデータの形式を解析することにより判別する。MIMEでは、データの形式はContent-Type:の後に記述され、イメージデータであればimage/fffと記され、文字コードデータであればtext/plainと記される。次にステップS22で、受信した文字コードデータを圧縮・伸長部8で伸長して外部記憶部4に蓄積した後、フォント部12でイメージデータに展開し、ステップS23で、展開したイメージデータを印刷する。なお、文字コードデータを送信する場合は、通常の電子メールと同様に行う。

【0026】このように、上記第3の実施例によれば、本装置を電子メール本来の装置としての使い方もでき、文字コードデータの電子メールを本装置に送信してもエラーとならない。

【0027】(実施例4) 次に、本発明の第4の実施例について図8を参照して説明する。本実施例は、図6に示した第3の実施例の構成に、文字認識部13を追加したものである。

【0028】図9は本実施例における宛先を文字認識する際のフローを示す。まずステップS31で原稿をスキャナ部6に乗せてスタートボタンを押す。次にステップS32で、スキャナ部6からイメージを入力する。次にステップS32で、入力されたイメージの所定の位置を解析し、宛先を文字認識部13で文字認識する。所定の位置の決め方は、本装置専用のカバーシートを作成し、宛先を書く位置をあらかじめ決めておく。そしてステッ

プS34で、認識した宛先に対してイメージデータを電子メールとして送信する。

【0029】このように、上記第4の実施例によれば、パネル部7からアルファベットのキー入力をしなくとも宛先アドレスを入力することができる。

【0030】(実施例5) 次に、本発明の第5の実施例について図6を参照して説明する。電子メール装置の構成は、図8に示した第4の実施例と同じである。

【0031】図10は本実施例における発信元のアドレスを入力する際のフローを示す。まずステップS41で、実施例1または4のように宛先アドレスを入力する。次にステップS42で、発信元ボタンを押し、発信元アドレスを入力する。次にステップS43で、スタートボタンを押し、以降のフローは図1と同様にしてイメージデータを送信する(ステップS44～S46)。発信元アドレスは、通常送信者の席にあるPCまたはWSの電子メールアドレスを入力する。したがって、電子メールが、もし送信失敗になれば、従来からある電子メール装置と同様に、その失敗通知メールを自席のPCまたはWSで受け取ることができる。

【0032】このように、上記第5の実施例によれば、送信が失敗した場合に、発信元アドレスを基に送信者本人に不達通知を届けけることができる。

【0033】(実施例6) 次に、本発明の第6の実施例について図11を参照して説明する。本実施例は、図8に示した第4の実施例の構成に、マイクとレシーバを有する音声入出力部14と、デジタル音声を圧縮または伸長する音声圧縮・伸長部15を追加したものである。

【0034】図12は本実施例における音声を入力してイメージと一緒に送信する際のフローを示す。まずステップS51で、実施例1または4のように宛先を入力し、ステップS51で、パネル部7にある伝言ボタンを押す。伝言ボタンを押しながら伝言を音声入出力部14のマイクを通じて録音する。音声は音声圧縮・伸長部15によって圧縮され、外部記憶部4に蓄積される。次にステップS53で、スタートボタンを押し、ステップS54で、スキャナ部6からイメージデータを入力し、ステップS55で、入力したイメージデータをフォーマット変換し、外部記憶部4に蓄積する。次にステップS56で、蓄積されたイメージデータと音声データの一つのメールに統合し、ステップS57で、宛先に対して送信する。受信する場合は、受信したイメージデータと音声データをそれぞれ圧縮・伸長部8、15で伸長し、プリンタ部11および音声入出力部14から出力する。なお、二つの異なったデータ形式のファイルを統合する方法は、上記したMIMEで規定されている。

【0035】このように、上記第6の実施例によれば、イメージ情報とともに、送信者の感情を伝える等の音声特有の情報を伝えることができる。

【0036】(実施例7) 次に、本発明の第7の実施例

について図13を参照して説明する。本実施例は、図11に示した第6の実施例の構成に、ビデオカメラとCRTを有する映像入出力部16と、動画圧縮・伸長部17を追加したものである。

【0037】図14は本実施例における動画を入力してイメージと一緒に送信する際のフローを示す。まずステップS61で、実施例1または4のように、宛先を入力し、ステップS62で、パネル部7の映像ボタンを押し、映像入出力部16のカメラで動画を録画する。録画された動画は、動画圧縮・伸長部17でMPEG方式で圧縮され、外部記憶部4に蓄積される。次にステップS63で、スタートボタンを押し、ステップS64で、スキャナ部6からイメージデータを入力し、ステップS65で、入力したイメージデータをフォーマット変換し、外部記憶部4に蓄積する。次にステップS66で、蓄積されたイメージデータと動画データを一つのメールに統合し、ステップS67で、宛先に対して送信する。受信する場合は、受信したイメージデータと動画データをそれぞれ圧縮・伸長部8、17で伸長し、プリンタ部11および映像入出力部16から出力する。なお、二つの異なったデータ形式のファイルを統合する方法は、上記したMIMEで規定されている。

【0038】このように、上記第7の実施例によれば、イメージ情報とともに、静止画では表現できない動画情報を伝えることができる。

【0039】(実施例8)次に、本発明の第8の実施例について説明する。本実施例では、図8に示した構成を用いて、LAN制御部9が受信した電子メールを解析し、CPU1が解析結果に応じた処理を行うようにしたものである。

【0040】図15は本実施例における電子メールにより宛先メールアドレスと発信元アドレスの登録をする際のフォーマット例を示す。電子メールは、大きくはヘッダと本文に別れており、ヘッダ部分で本文のデータ形式が解る。図15のヘッダ31のContent-type:text/plainとあり、これは本文32が文字コードデータであることを示しており、本実施例では、本文が文字コードデータであることが必要である。ヘッダ31のSubject:!!\$は、本文32が命令であることを示している。Subject:の後ろには、通常この電子メールの題名が入るので、!!\$は特別な文字コードであり、題名とは区別できる。

【0041】本文32には、送信者ID、宛先メールアドレスのリスト、発信元アドレス等が記され、それぞれの先頭には、!ID,!LIST,!FROMと定義されている。本装置は、LAN制御部9が、受信した文字コードデータの所位置にあるこれらの文字列を解析し、その定義されている文字コードから送信者ID等を識別し、それらをCPU1が外部記憶部4に登録するようにしたものである。

【0042】このように、上記第8の実施例によれば、

本装置とPCやWS等の端末との間に特別なプロトコルの必要なしに、必要な処理を行うことができる。

【0043】(実施例9)次に、本発明の第9の実施例について説明する。本実施例は、図8に示した構成を用いて、LAN制御部9が、受信した文字コードデータの電子メールを解析し、CPU1が、送信者IDと宛先メールアドレスを対応させて登録するようにしたものである。

【0044】図16は本実施例のフローを示す。まずステップS71で、LAN制御部9が、電子メールの文字コードデータを受信すると、ステップS72で、ヘッダのSubject部を解析する。次にステップS73で、解析結果が本文の命令であるかどうかを判断し、解析結果の本文の命令であることが判明すれば、ステップS74で、本文を解析する。そしてステップS75で、CPU1が、送信者IDと宛先メールアドレスを対応させて外部記憶部4に登録する。本文が命令でない場合は、ステップS76で、文字データをイメージデータに変換し、ステップS77で、そのイメージデータを印刷する。

【0045】図17は本実施例における宛先リストから宛先を選ぶ際のフローを示す。まずステップS81で、パネル部7の宛先リストボタンを押し、ステップS82で、送信者IDを入力すると、電子メールで登録した宛先リストがパネル部7に表示される。そしてステップS83で、表示された宛先リストから宛先を選び、ステップS84で、スタートボタンを押すと、スキャナ部6から入力されたイメージデータが、フォーマット変換されてその宛先に送信される(ステップS85～S87)。

【0046】このように、上記第9の実施例によれば、宛先電子メールアドレス一覧を個人毎に簡単に入力することができる。

【0047】(実施例10)次に、本発明の第10の実施例について説明する。本実施例は、図8に示した構成を用いて、LAN制御部9が、受信した文字コードデータの電子メールを解析し、CPU1が、送信者IDと発信元アドレスを対応させて外部記憶部4に登録するようにしたものであり、上記第9の実施例と異なるのは、図16において、ステップS75で送信者IDと発信元アドレスを対応させて登録することである。発信元アドレスを選ぶ時は、パネル部7の発信元ボタンを押し、送信者IDを入力すると、登録した発信元アドレスがパネル部7に表示されるので、それを見ながら入力し、スタートボタンを押す。

【0048】このように、上記第10の実施例によれば、送信者(発信元)の電子メールアドレスを個人毎に簡単に入力することができる。

【0049】(実施例11)次に、本発明の第11の実施例について説明する。本実施例は、上記第9の実施例と第10の実施例とを組み合わせたものであり、さらに、宛先リストから宛先を選んだときに発信元アドレス

が決定されるようにしたものである。すなわち、図17において、ステップS82で送信者IDを入力したときに、対応する発信元アドレスが認識され、イメージデータを送信するときに、ヘッダにその発信元アドレスが自動的に記入される。

【0050】このように、上記第11の実施例によれば、発信元の電子メールアドレスの入力を送信者は意識することなく行うことができる。

【0051】(実施例12)次に、本発明の第12の実施例について図18を参照して説明する。本実施例は、図12に示した構成にファクシミリの手順でイメージを送るためのモデム部18を追加したものである。モデム部18にはハンドセットが付属しているものとする。

【0052】図19は本実施例における電子メールとファクシミリのいずれかを選択して送信する際のフローを示す。まずステップS91で、パネル部7から電子メールかファクシミリかを選択するボタンを押す。次にステップS92で、ファクシミリと判定されれば、ステップS93で宛先番号を入力してスタートボタンを押す。ステップS94で、スキャナ部6からイメージを入力して、ステップS95で、宛先に対してファクシミリの手順で送信する。ステップS92で、ファクシミリでないと判定されれば、電子メールとして図2と同じ手順で送信する(ステップS96～S99)。

【0053】このように、上記第12の実施例によれば、電話回線を通じてファクシミリ送信ができるので、相手が電子メールを受信できない場合には、ファクシミリによりイメージデータを送信することができる。

【0054】(実施例13)次に、本発明の第13の実施例について説明する。本実施例は、図18に示した構成を用いて、受信待ちの端末に受信通知をメールするようにしたものであり、そのフローを図20に示す。まず図5とおなじ手順でイメージを印刷した後(ステップS101～S103)、ステップS104で、受信待ちフラグがあると判定されれば、ステップS105で、受信通知をあらかじめ登録された端末に対して送信する。

【0055】受信通知フラグを立てるためには、図15のフォーマットに従って、本文に!RCV SETを記入した電子メールをあらかじめ送信しておく。また、受信通知フラグをリセットするには、!RCV RESETを本文に記入した電子メールを送る。

【0056】このように、上記第13の実施例によれば、ファクシミリデータを受信したことを自己の端末で知ることができる。

【0057】(実施例14)次に、本発明の第14の実施例について説明する。実施例は、図18に示した構成を用いて、受信した電子メールをファクシミリ手順で送信するようにしたものであり、そのフローを図21に示す。まずステップS110で、イメージデータを受信後、ステップS111で、イメージデータを逆フォーマ

ット変換し、ステップS112で、図15のフォーマットに従ってヘッダの解析を行い、Subject にイメージをFAX送信する意味の文字列があれば、その後に続く文字列を宛先のFAX番号と解釈する。そしてステップS113で、FAX送信と判断されれば、ステップS114で、述のFAX番号に対してイメージデータをFAX送信する。FAX送信でなければ、ステップS115で、印刷する。

【0058】このように、上記第14の実施例によれば、個人の端末からイメージデータをファクシミリに送信することができる。

【0059】(実施例15)次に、本発明の第15の実施例について図22を参照して説明する。本実施例は、図18に示した構成に世界の現地時間にイメージを送るための世界時計部19を追加したものであり、この世界時計部19は、国名から現地の時間を得ることができる時計である。

【0060】図23は本実施例における宛先の国の時刻でタイマー送信する際にフローを示す。通常の実送と異なるのは、ステップS121で宛先を入力した後、ステップS122で、送信先の時刻で送信時刻をセットすることと、スタートボタンを押して入力したイメージデータをフォーマット変更した後(ステップS123～S125)、ステップS126で、宛先から国を判別することと、ステップS127で、宛先と世界時計部19からその国の現地時刻を得、その時刻とセットされた時刻が合ったときにイメージデータを送信することである。電子メールの宛先は、例えば、loyo@mgcs.mei.co.jpのようになり、最後のjpが日本を表している。このように最後の文字列で国が分かる仕組みになっている。

【0061】このように、上記第15の実施例によれば、送信者が世界各国の現地時間を計算する必要なく、現地時間でタイマ送信することができる。

【0062】(実施例16)次に、本発明の第16の実施例について説明する。本実施例は、図18に示した構成を用いて、自装置に複数の電子メールアドレスを持つようにしたものであり、そのフローを図24に示す。イメージデータを受信して逆フォーマット変換した後(ステップS131、S132)、ステップS133で、外部記憶部4に作成された宛先毎のboxにイメージデータを蓄積する。このboxは、所有者があらかじめ入力したID、パスワードと対応付けされており、boxの内容を取り出すためには宛先毎の所有者のIDとパスワードが必要である。そしてステップS134で、所有者のIDとパスワードを入力し、これが認められれば、ステップS135でその内容を印刷する。これらの処理は、ROM2に格納されたプログラムに基づいて実行される。

【0063】このように、上記第16の実施例によれば、電子メールを受信する端末を持たない人でも受信デ

ータを個人毎に管理できる。

【0064】（実施例17）次に、本発明の第17の実施例について説明する。本実施例は、図18に示した構成を用いて、受信側でカバーシートを付加するようにしたものであり、そのフローは、図24に対して、ステップS134が無くなり、ステップS135で所有者毎のカバーシートを付加するものである。カバーシートは、所有者によって、あらかじめ外部記憶部4の所有者のboxにスキャナ部6から読み込むか、またはカバーシートを登録する命令を含んだ電子メールを本装置に送信して登録しておく。

【0065】このように、上記第17の実施例によれば、本装置への送信者がカバーシートを付けなくとも誰宛の受信データかがわかる。

【0066】（実施例18）次に、本発明の第18の実施例について説明する。本実施例は、図18に示した構成を用いて、送信者に本装置が受信した旨の確認メールを返信するようにしたものであり、図25にそのフローを示す。イメージデータを受信し、逆フォーマットに変換して印刷した後（ステップS141～S143）、ステップS144で、ヘッダの送信者の電子メールアドレスへ受信した旨の確認のメールを送信する。確認のメールには、受信した旨の文字列と本装置のメールアドレスおよび会社名等を記載する。これらの処理は、ROM2に格納されたプログラムに基づいて実行される。

【0067】このように、上記第18の実施例によれば、本装置への送信者が本装置に電子メールが届いたことがわかる。

【0068】（実施例19）次に、本発明の第19の実施例について説明する。実施例は、図18に示した構成を用いて、送信者に受信者がデータを受け取った旨の確認メールを送信するようにしたものであり、図26にそのフローを示す。イメージデータを受信して逆フォーマット変換した後（ステップS151、S152）、ステップS153で、イメージデータに受付IDを記載する。受付IDは、一年間ユニークになるような日付、時間を基本にして生成する。次にステップS154で、受信者がイメージデータに記載された受付IDを入力すると、ステップS155で、受付IDに対応したイメージデータの送信者のアドレスをヘッダから解釈し、確認のメールを送信者宛に送信する。

【0069】このように、上記第19の実施例によれば、本装置への送信者が目的とする個人に電子メールが届いたことがわかる。

【0070】（実施例20）次に、本発明の第20の実施例について説明する。実施例は、図18に示した構成を用いて、送信失敗の旨を印刷するようにしたものであり、図27にそのフローを示す。ステップS161で、文字コードデータを受信し、ステップS162で、受信メールのヘッダのFrom行、即ち送信者のアドレスを解析

し、ステップS163で自装置のアドレスであると判断された場合は、ステップS164で、送信失敗の旨を印刷する。自装置以外のメールであれば、ステップS165で、文字データをイメージデータに変換し、ステップS166で、その内容を印刷する。

【0071】このように、上記第20の実施例によれば、本装置から送信したイメージデータがエラーになったことが、電子メールを扱ったことのない人でも容易にわかる。

10 【0072】なお、上記第2から第20までの実施例において、電子メールで受信した文字コードをCRT等のディスプレイ上に表示するようにしてもよく、動画を表示する場合も、このCRTを利用することができる。また、上記第1から第20までの実施例を種々に組み合わせて本装置を構成することができる。

【0073】

【発明の効果】本発明による電子メール装置は、上記各実施例から明らかなように、原稿を読み取ったイメージデータを電子メールのフォーマットに変換してからネットワーク上に送信し、受信する際は、逆に電子フォーマットをファクシミリのフォーマットに変換した後に、プリンタで印刷するようにしたので、本装置を電子メールが可能なネットワークに接続することにより、従来のファクシミリ装置のように、単に原稿を読み取らせるという簡単な操作だけで、イメージデータを相手に送信することができ、受信する際は、単に装置を動作状態にしておくだけで、イメージデータを受信することができる。しかも、従来のファクシミリ装置のように送信相手に確実に届いたかどうか分らなかったり、他人に読まれたり、通信エラーが発生するというような問題を解決できる。

30 【0074】本発明はまた、ネットワークを通じた電子メールによるイメージデータの送受信と、電話回線を通じたファクシミリによるイメージデータの送受信とをいづれも可能にしたので、送信相手が電子メール装置である場合は電子メールにより、また送信相手がファクシミリ装置の場合はファクシミリ装置により送信することができる。さらに、受信した電子メールをファクシミリ装置で送信したり、受信したファクシミリを電子メールで受信したりすることができる。

【図面の簡単な説明】

【図1】本発明の第1の実施例における電子メール装置の概略ブロック図

【図2】本発明の第1の実施例におけるイメージデータを送信する際のフロー図

【図3】本発明の第1の実施例におけるイメージデータの電子メールフォーマットへの変換方法を示すブロック図

【図4】本発明の第2の実施例における電子メール装置の概略ブロック図

【図5】本発明の第2の実施例におけるイメージデータを受信する際のフロー図

【図6】本発明の第3の実施例における電子メール装置の概略ブロック図

【図7】本発明の第3の実施例における文字コードデータを印刷する際のフロー図

【図8】本発明の第4、第8、第9、第10、第11の実施例における電子メール装置の概略ブロック図

【図9】本発明の第4の実施例における宛先を文字認識する際のフロー図

【図10】本発明の第5の実施例における発信元のアドレスを入力する際のフロー図

【図11】本発明の第6の実施例における電子メール装置の概略ブロック図

【図12】本発明の第6の実施例における音声を入力してイメージと一緒に送信する際のフロー図

【図13】本発明の第7の実施例における電子メール装置の概略ブロック図

【図14】本発明の第7の実施例における動画を入力してイメージと一緒に送信する際のフロー図

【図15】本発明の第8の実施例における電子メールアドレスによる宛先メールアドレスと発信元アドレスの登録例を示すデータ構成図

【図16】本発明の第9、第10、第11の実施例における受信した文字コードデータの電子メールを解析する際のフロー図

【図17】本発明の第9、第10、第11の実施例における宛先リストから宛先を選ぶ際のフロー図

【図18】本発明の第12、第13、第14、第16、第17、第18、第19、第20の実施例における電子メール装置の概略ブロック図

【図19】本発明の第12の実施例における電子メールとファクシミリを選択して送信する際のフロー図

【図20】本発明の第13の実施例における受信待ちをしている端末に受信通知をメールする際のフロー図

【図21】本発明の第14の実施例における受信したメールをファクシミリ送信する際のフロー図

【図22】本発明の第15の実施例における電子メール装置の概略ブロック図

【図23】本発明の第15の実施例における宛先の国の時刻でタイマー送信する際のフロー図

【図24】本発明の第16、第17の実施例における自装置に複数の電子メールアドレスを持っているときのイメージデータを受信する際のフロー図

【図25】本発明の第18の実施例における送信者に受信した旨の確認メールを返信する際のフロー図

10 【図26】本発明の第18の実施例における送信者に受信者がデータを受け取った旨の確認メールを返信する際のフロー図

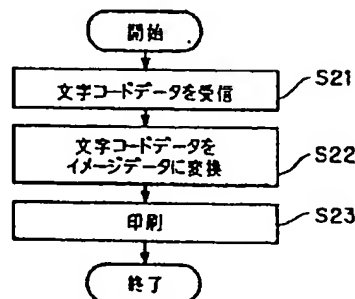
【図27】本発明の第20の実施例における送信失敗の旨を印刷する際のフロー図

【図28】従来の電子メールを利用したファクシミリ装置の概略ブロック図

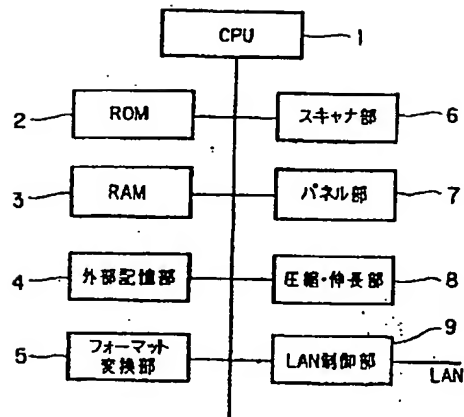
【符号の説明】

- |    |            |
|----|------------|
| 1  | CPU        |
| 2  | ROM        |
| 20 | 3 RAM      |
| 4  | 外部記憶部      |
| 5  | フォーマット変換部  |
| 6  | スキャナ部      |
| 7  | パネル部       |
| 8  | 圧縮・伸長部     |
| 9  | LAN制御部     |
| 10 | フォーマット逆変換部 |
| 11 | プリンタ部      |
| 12 | フォント部      |
| 30 | 13 文字認識部   |
| 14 | 音声入出力部     |
| 15 | 音声圧縮・伸長部   |
| 16 | 映像入出力部     |
| 17 | 動画圧縮・伸長部   |
| 18 | モデム部       |
| 19 | 世界時計部      |

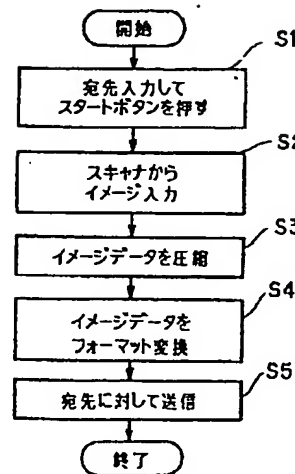
【図7】



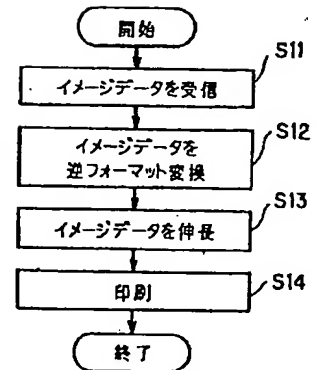
【図1】



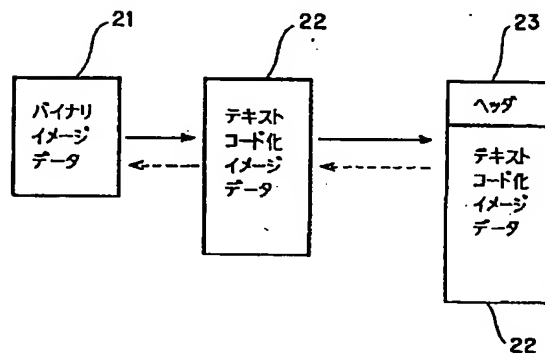
【図2】



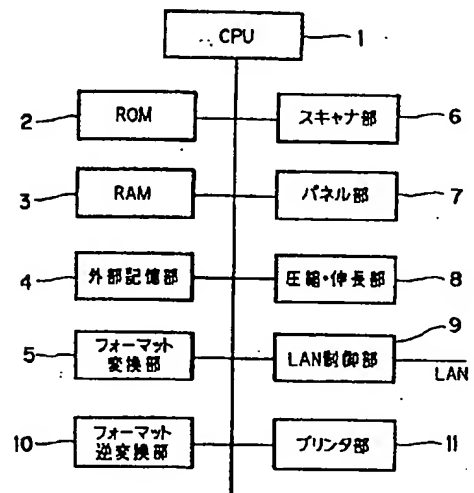
【図5】



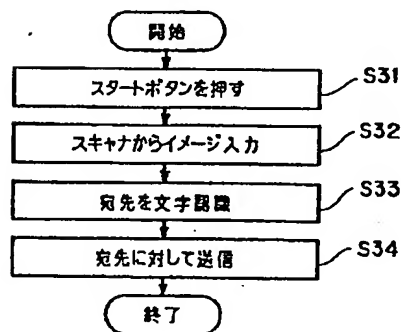
【図3】



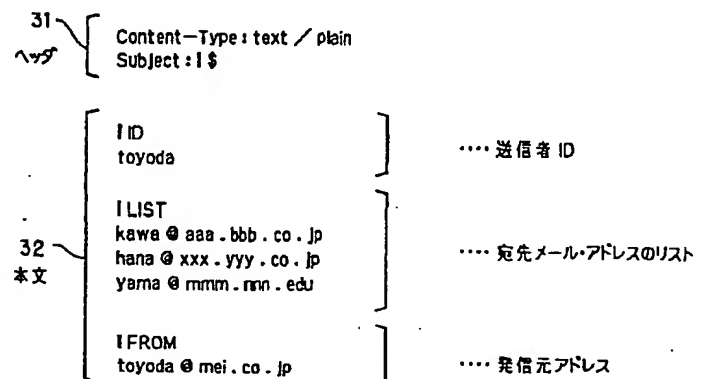
【図4】



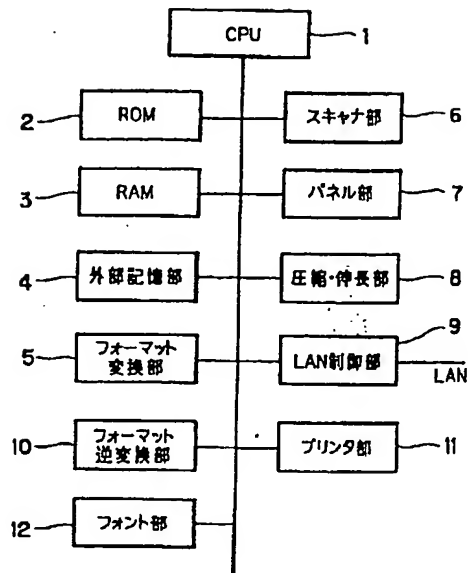
【図9】



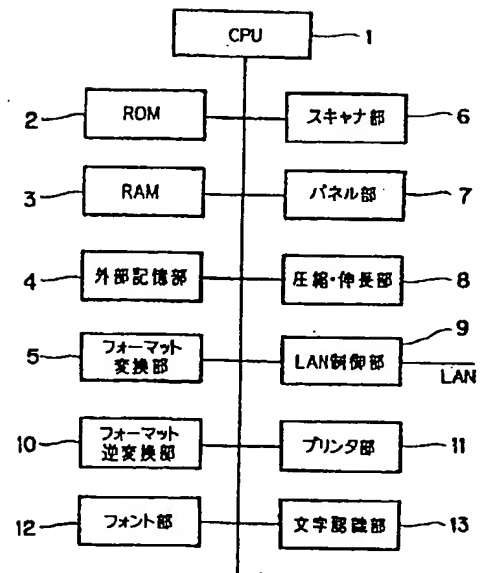
【図15】



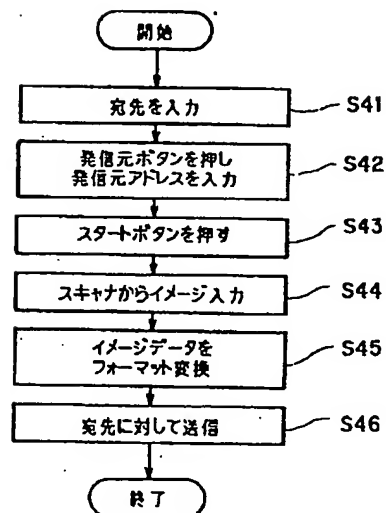
【図6】



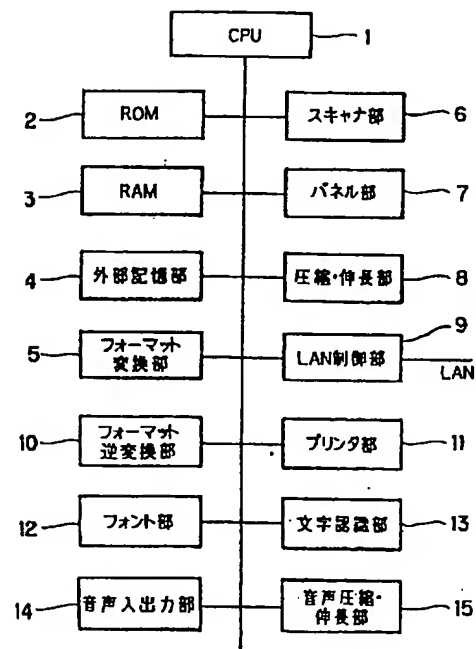
【図8】



【図10】

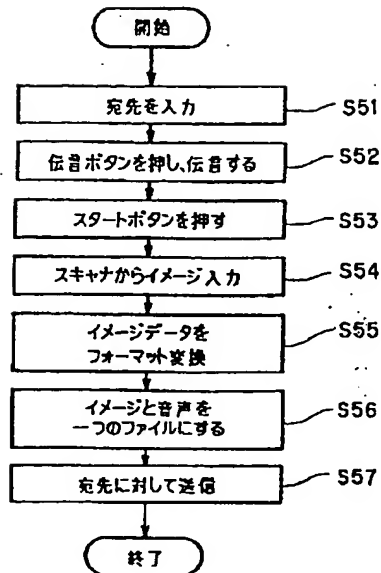


【図11】

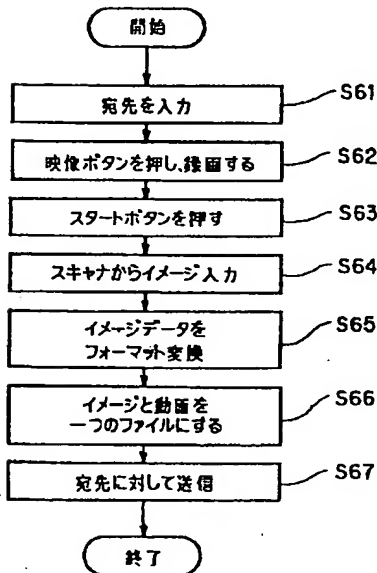




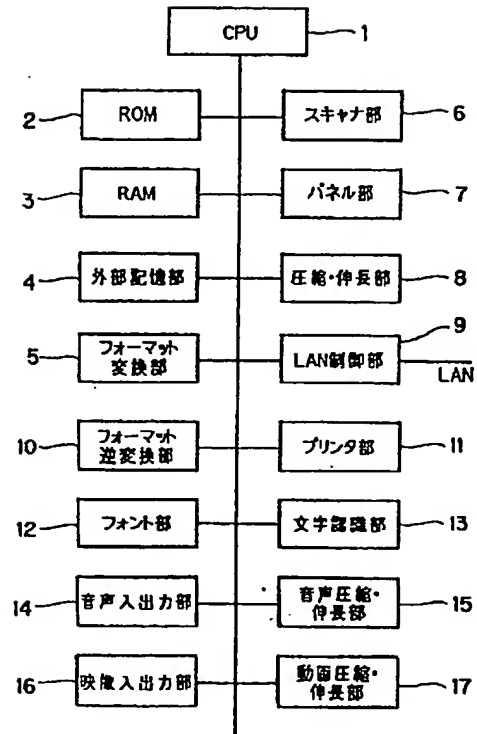
【図12】



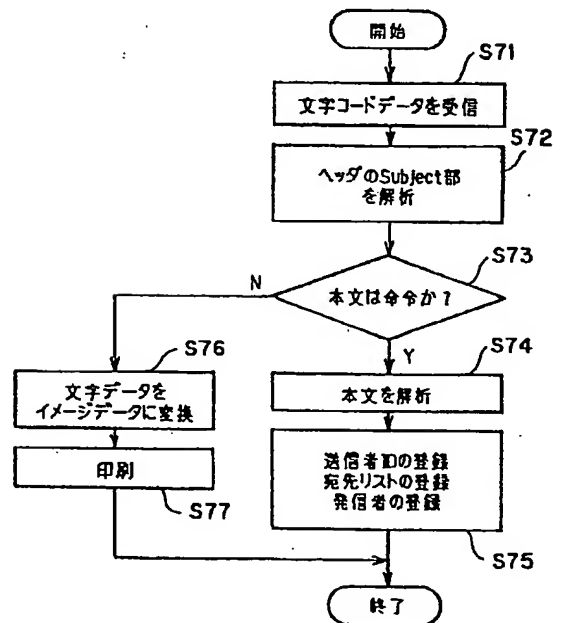
【図14】



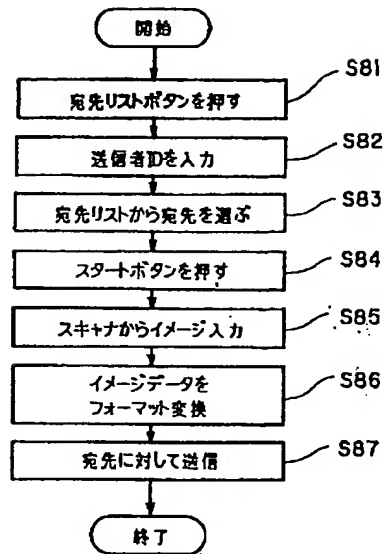
【図13】



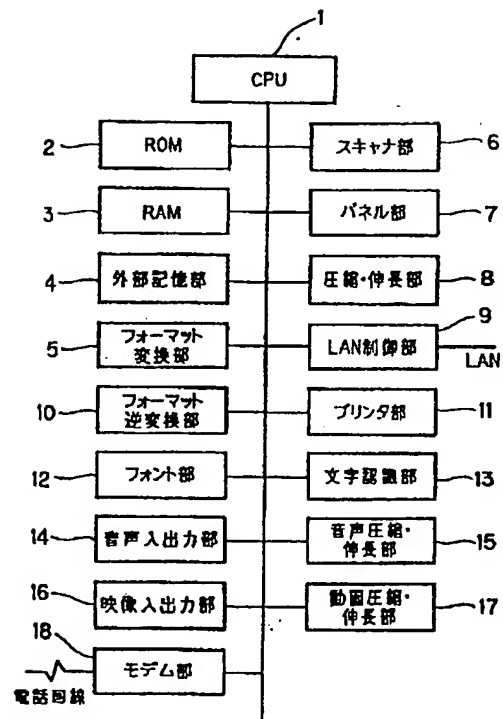
【図16】



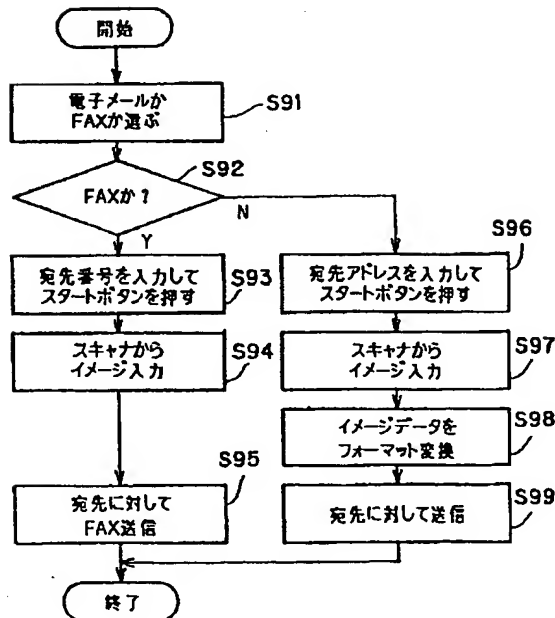
【図17】



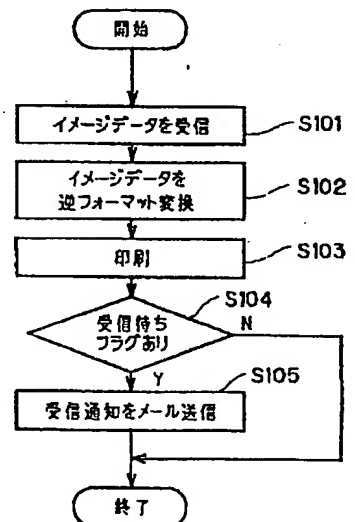
【図18】



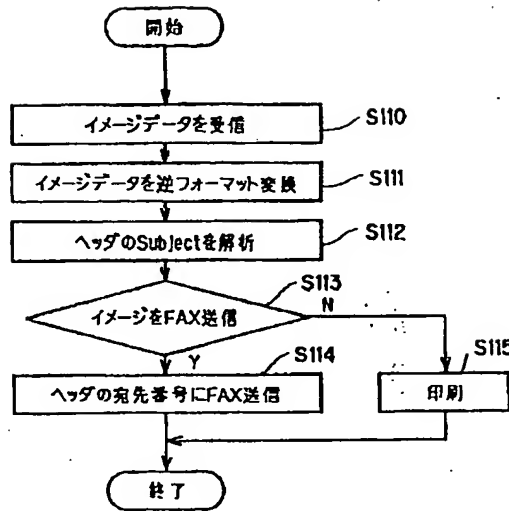
【図19】



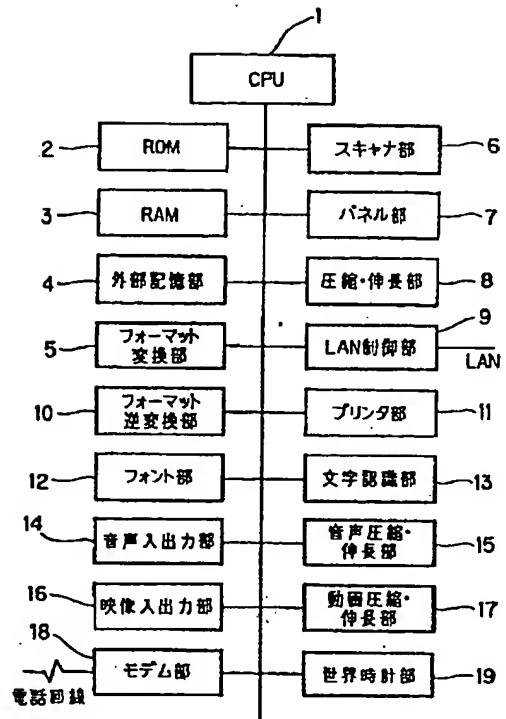
【図20】



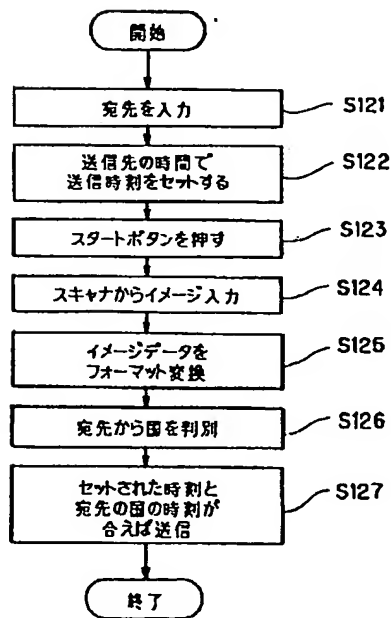
【図 2 1】



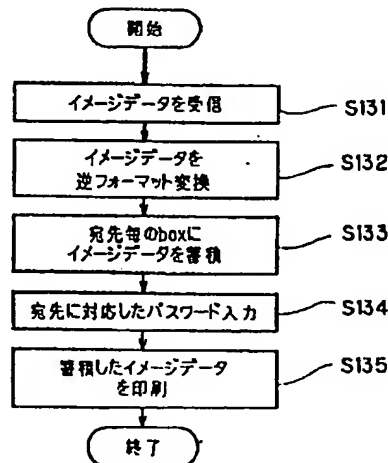
【図 2 2】



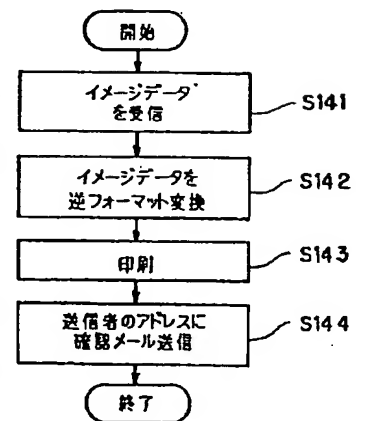
【図 2 3】



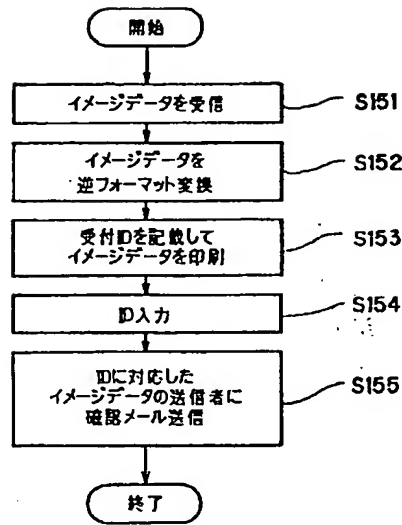
【図 2 4】



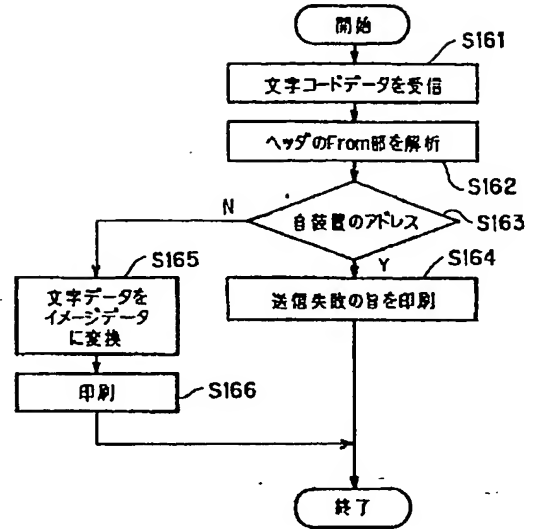
【図 2 5】



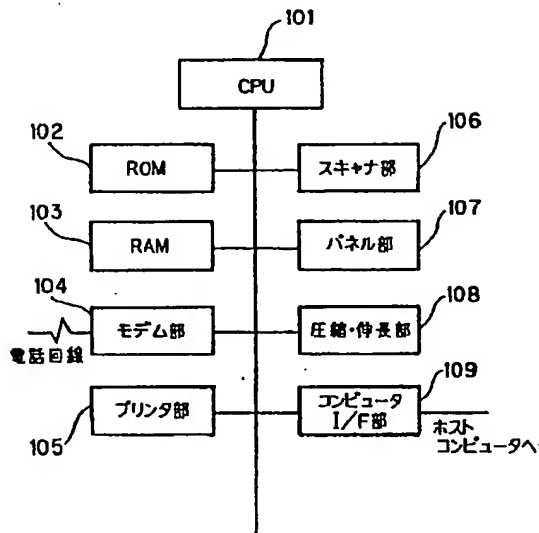
【図26】



【図27】



【図28】



# 甲 8 号 証

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-107944

(43) 公開日 平成10年(1998) 4月24日

(51) Int.Cl. <sup>4</sup>	識別記号	FI
H04N 1/00	107	H04N 1/00 107Z
H04L 12/54		H04M 11/00 303
12/58		H04N 1/32 F
H04M 11/00	303	H04L 11/20 101C
H04N 1/32		101B

審査請求 未請求 請求項の数7 OL (全13頁)

(21) 出願番号 特願平8-262627

(22) 出願日 平成8年(1996)10月3日

(71) 出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72) 発明者 太田 直樹

東京都大田区中馬込1丁目3番6号 株式会社リコー内

(72) 発明者 大瀬戸 太

東京都大田区中馬込1丁目3番6号 株式会社リコー内

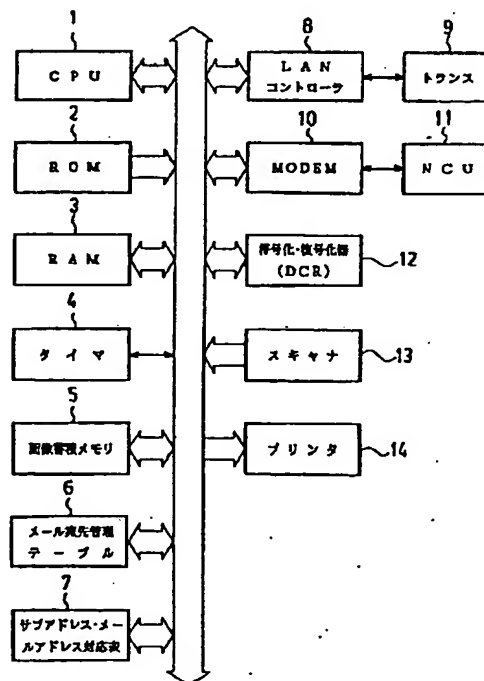
(74) 代理人 弁理士 大澤 敬

(54) 【発明の名称】 ファクシミリ装置

(57) 【要約】

【課題】 ネットワーク上のサーバに対して電子メール化した同じ内容のFAX文書を一度に複数転送しなくても各クライアント端末装置が自装置宛の電子メールを滞り無く受け取れるようにする。

【解決手段】 CPU1は、公衆電話網からFAX文書を受信すると、画像蓄積メモリ5に画像データを蓄積し、その宛先にFAX文書の着信を知らせる着信通知メールを作成してネットワーク上のメールサーバへ送信して、そのメールサーバを介して着信通知メールの宛先から画像要求メールを受信すると、画像蓄積メモリ5から該当するFAX文書の画像データを読み出し、その電子メールを作成してメールサーバへ送信する。



## 【特許請求の範囲】

【請求項1】 インターネット、イーサネット等のネットワークとの電子メールの通信制御を行なう手段と、公衆電話網との画像データのファクシミリ通信制御を行なう手段と、前記公衆電話網から画像データとその宛先のアドレス情報とを受信して前記画像データを蓄積する手段と、

前記宛先のアドレス情報に基づいてその宛先に画像データの着信を知らせる着信通知メールを作成する手段と、該手段によって作成された着信通知メールを前記ネットワーク上のサーバへ送信する手段と、

前記サーバを介して前記着信通知メールの宛先から画像要求メールを受信する手段と、該手段によって受信した画像要求メールに基づいて前記蓄積した中から該当する画像データの電子メールを作成する手段と、該手段によって作成された電子メールを前記サーバへ送信する手段とを備えたことを特徴とするファクシミリ装置。

【請求項2】 請求項1記載のファクシミリ装置において、

前記公衆電話網から受信した画像データの宛先毎に画像要求メールの着信の有無を管理する宛先管理テーブルと、該宛先管理テーブルに基づいて全宛先から画像要求メールを受信してそれぞれに画像データの電子メールを配信済みにしたと判断した後、前記蓄積した中から該当する画像データを削除する手段とを設けたことを特徴とするファクシミリ装置。

【請求項3】 請求項1記載のファクシミリ装置において、

前記画像データに設定された保持時間をカウントするタイマと、該タイマによって保持時間が経過したとき、前記蓄積した中から該当する画像データを削除する手段とを設けたことを特徴とするファクシミリ装置。

【請求項4】 請求項1記載のファクシミリ装置において、

前記着信通知メールに画像データが削除される予定時間を付加する手段と、前記画像データに設定された保持時間をカウントするタイマと、該タイマによって保持時間が経過したとき、前記画像要求メールの未受信の宛先へ画像データが削除されることを知らせる警告メールを作成して送信する手段と、該手段によって警告メールを送信した後にさらに所定時間を経過したとき、前記蓄積した中から該当する画像データを削除する手段とを設けたことを特徴とするファクシミリ装置。

【請求項5】 請求項1記載のファクシミリ装置において、

前記着信通知メールに画像データが削除される予定時間を付加する手段と、前記画像データに設定された保持時間をカウントするタイマと、前記タイマによって保持時間が経過したときに前記画像要求メールの未受信の宛先があったとき、前記蓄積した中から該当する画像データ

をプリント出力した後に削除する手段とを設けたことを特徴とするファクシミリ装置。

【請求項6】 請求項1記載のファクシミリ装置において、

前記画像データに設定された保持時間をカウントするタイマと、該タイマによって保持時間が経過したとき、前記画像データに複数の個人宛が指定されていたとき、前記画像要求メールの未受信の宛先に対して前記蓄積した中から該当する画像データを一定時間間隔でそれぞれ送信した後に前記該当する画像データを削除する手段とを設けたことを特徴とするファクシミリ装置。

【請求項7】 請求項6記載のファクシミリ装置において、

前記宛先から画像データの電子メールの送信不要を指定する電子メールを受信したとき、その宛先への電子メールの送信を実行せずに送信済みとして処理する手段を設けたことを特徴とするファクシミリ装置。

## 【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、ファクシミリ文書を受受信すると共に、そのファクシミリ文書をインターネット、イーサネット等のネットワークへ電子メールとして送信するファクシミリ装置に関する。

【0002】

【従来の技術】従来、公衆電話網(PSTN)を介してファクシミリ文書(FAX文書)を受信し、そのFAX文書のサブアドレス等の情報に基づいて電子メールアドレスを作成し、その電子メールアドレスに上記FAX文書を付加して電子メール化して、インターネット、イーサネット(Ethernet)等のネットワーク上のサーバに設けられた上記電子メールアドレスに対応するメールボックスに(Mailbox)電子メール化したFAX文書を転送するFAX-Mailゲートウェイ等のネットワーク対応型ファクシミリ装置(例えば、特開平6-164645号公報、特開平6-261164号公報参照)があった。

【0003】そして、ネットワーク上の各クライアント端末装置のユーザは、サーバ上の自己のクライアント端末装置の電子メールアドレスに対応するメールボックスに配信された電子メール化されたFAX文書を取り出して参照することができる。

【0004】なお、上記インターネットとは、多数のサーバによって接続された多数のネットワークからなる世界規模の通信ネットワークであり、電子メールやワールドワイドウェブ(WWW)等のサービス機能を利用できる。

【0005】このようなファクシミリ装置では、受信したFAX文書のサブアドレス等の情報が複数の宛先を指定している場合、その指定された各宛先について上記受信したFAX文書を複製して電子メールを作成し、ネッ

トワーク上のサーバの各電子メールの電子メールアドレスに対応するメールボックスに転送していた。

【0006】

【発明が解決しようとする課題】しかしながら、上述した従来のファクシミリ装置では、FAX受信したFAX文書を転送先の宛先は異なるが同じ内容の複数の電子メールをネットワーク上のサーバに送信してしまうので、サーバの処理負担を増加させてしまうという問題があった。

【0007】特に、電子メール化したFAX画像は通常のキャラクタデータのための電子メールと異なってデータサイズが大容量になるので、このような電子メールをサーバへ一度に多数通送すると、サーバの負荷が大きくなってネットワーク上の電子メールのやり取りが滞ってしまうという問題があった。

【0008】この発明は上記の点に鑑みてなされたものであり、ネットワーク上のサーバに対して電子メール化した同じ内容のFAX文書を一度に複数転送しなくても各クライアント端末装置が自装置宛の電子メールを滞り無く受け取れるようにすることを目的とする。

【0009】

【課題を解決するための手段】この発明は上記の目的を達成するため、インターネット、イーサネット等のネットワークとの電子メールの通信制御を行なう手段と、公衆電話網との画像データのファクシミリ通信制御を行なう手段と、上記公衆電話網から画像データとその宛先のアドレス情報とを受信して上記画像データを蓄積する手段と、上記宛先のアドレス情報に基づいてその宛先に画像データの着信を知らせる着信通知メールを作成する手段と、その手段によって作成された着信通知メールを上記ネットワーク上のサーバへ送信する手段と、上記サーバを介して上記着信通知メールの宛先から画像要求メールを受信する手段と、その手段によって受信した画像要求メールに基づいて上記蓄積した中から該当する画像データの電子メールを作成する手段と、その手段によって作成された電子メールを上記サーバへ送信する手段を備えたファクシミリ装置を提供する。

【0010】また、上記公衆電話網から受信した画像データの宛先毎に画像要求メールの着信の有無を管理する宛先管理テーブルと、その宛先管理テーブルに基づいて全宛先から画像要求メールを受信してそれぞれに画像データの電子メールを配信済みにしたと判断した後、上記蓄積した中から該当する画像データを削除する手段を設けるとよい。

【0011】さらに、上記画像データに設定された保持時間をカウントするタイマと、そのタイマによって保持時間が経過したとき、上記蓄積した中から該当する画像データを削除する手段を設けるとよい。

【0012】また、上記着信通知メールに画像データが削除される予定時間を付加する手段と、上記画像データ

に設定された保持時間をカウントするタイマと、そのタイマによって保持時間が経過したとき、上記画像要求メールの未受信の宛先へ画像データが削除されることを知らせる警告メールを作成して送信する手段と、その手段によって警告メールを送信した後にさらに所定時間を経過したとき、上記蓄積した中から該当する画像データを削除する手段を設けるとよい。

【0013】さらに、上記着信通知メールに画像データが削除される予定時間を付加する手段と、上記画像データに設定された保持時間をカウントするタイマと、上記タイマによって保持時間が経過したときに上記画像要求メールの未受信の宛先があったとき、上記蓄積した中から該当する画像データをプリント出力した後に削除する手段を設けるとよい。

【0014】また、上記画像データに設定された保持時間をカウントするタイマと、そのタイマによって保持時間が経過したとき、上記画像データに複数の個人宛が指定されていたとき、上記画像要求メールの未受信の宛先に対して上記蓄積した中から該当する画像データを一定時間間隔でそれぞれ送信した後に上記該当する画像データを削除する手段を設けるとよい。

【0015】さらに、上記宛先から画像データの電子メールの送信不要を指定する電子メールを受信したとき、その宛先への電子メールの送信を実行せずに送信済みとして処理する手段を設けるとよい。

【0016】この発明の請求項1のファクシミリ装置は、公衆電話網から画像データとその宛先のアドレス情報を受信し、その画像データを蓄積し、その宛先のアドレス情報に基づいてその宛先に画像データの着信を知らせる着信通知メールを作成してインターネット、イーサネット等のネットワーク上のサーバへ送信し、サーバを介して着信通知メールの宛先から画像要求メールを受信すると、その受信した画像要求メールに基づいて蓄積した中から該当する画像データの電子メールを作成してサーバへ送信する。

【0017】したがって、FAX文書をインターネット、イーサネット等のネットワーク上のサーバを介して電子メールとしてネットワーク上の各クライアント端末装置へ配信するとき、FAX文書の着信通知に対して要求があったときにのみそのクライアント端末装置へFAX文書を転送するので、ネットワーク上のサーバに電子メール化した同じ内容のFAX文書を一度に貯め込むことなく、各クライアント端末装置が自装置宛の電子メールを滞り無く受け取れる。

【0018】また、この発明の請求項2のファクシミリ装置は、公衆電話網から受信した画像データの宛先毎に画像要求メールの着信の有無を管理する宛先管理テーブルに基づいて、全宛先から画像要求メールを受信してそれぞれに画像データの電子メールを配信済みにしたと判断した後、蓄積した中から該当する画像データを削除す

る。

【0019】したがって、受信したFAX文書の宛先への配信を済ませたことを確認した後に速やかにそのFAX文書の画像データを削除することができるので、画像メモリ内に不要な画像データを長時間留めておくことがなくなり、メモリ容量を有効に活用することができる。

【0020】さらに、この発明の請求項3のファクシミリ装置は、画像データに設定された保持時間をカウントするタイマによって保持時間が経過したとき、蓄積した中から該当する画像データを削除する。したがって、受信したFAX文書の画像データを予め設定した有効期限の経過後に削除することができるので、画像メモリ内に画像データを長時間留めておくことがなくなり、メモリ容量を有効に活用することができる。

【0021】また、この発明の請求項4のファクシミリ装置は、着信通知メールに画像データが削除される予定時間を付加し、画像データに設定された保持時間をカウントするタイマによって保持時間が経過したとき、画像要求メールの未受信の宛先へ画像データが削除されることを知らせる警告メールを作成して送信し、その警告メールを送信した後にさらに所定時間を経過したとき、蓄積した中から該当する画像データを削除する。

【0022】したがって、受信したFAX文書の画像データに対して予め設定した削除期限を配信先のクライアント端末装置に知らせることができるので、各クライアント端末装置に対して必要なFAX文書の受け取り忘れを防止することができる。

【0023】さらに、この発明の請求項5のファクシミリ装置は、着信通知メールに画像データが削除される予定時間を付加し、画像データに設定された保持時間をカウントするタイマによって保持時間が経過したときに画像要求メールの未受信の宛先があったとき、蓄積した中から該当する画像データをプリント出力した後に削除する。

【0024】したがって、受信したFAX文書の画像データに対して予め設定した削除期限に削除する前にその画像データをプリントして残しておくことができるので、各クライアント端末装置がFAX文書の受け取りをうっかり忘れてしまってもその内容を提供することができる。

【0025】また、この発明の請求項6のファクシミリ装置は、画像データに設定された保持時間をカウントするタイマによって保持時間が経過したとき、画像データに複数の個人宛が指定されていたとき、画像要求メールの未受信の宛先に対して蓄積した中から該当する画像データを一定時間間隔でそれぞれ送信した後に該当する画像データを削除する。

【0026】したがって、所定時間内に受け取り要求がなかった複数のクライアント端末装置に対して一定時間

毎にそれぞれFAX文書の電子メールを送信して削除するので、ネットワーク上のサーバに負荷をかけることがなくなり、画像メモリ内に画像データを長期間留めておくことがなくなる。

【0027】さらに、この発明の請求項7のファクシミリ装置は、着信通知メールを送信した宛先から画像データの電子メールの送信不要を指定する電子メールを受信したとき、その宛先への電子メールの送信を実行せずに送信済みとして処理する。したがって、クライアント端末装置で必要としないFAX文書の配布をせずに済む。

【0028】

【発明の実施の形態】以下、この発明の実施の形態を図面に基づいて具体的に説明する。図1はこの発明のファクシミリ装置の一実施形態であるファックスメールゲートウェイ(FAX-Mail Gateway、以下「FAX-MG」と略称する)のハードウェア構成を示すブロック図である。

【0029】このFAX-MGは、CPU1、ROM2、RAM3、タイマ4、画像蓄積メモリ5、メール宛先管理テーブル6、サブアドレス・メールアドレス対応表7、LANコントローラ8、トランス9、モデム(MODEM)10、ネットワークコントロールユニット(NCU)11、符号化・復号化器(DCR)12、スキャナ13、及びプリンタ14等からなる。

【0030】CPU1はこのファクシミリ装置全体の制御を司り、この発明に関わるFAX文書を電子メールで配布するファクシミリ制御等を行なう中央処理装置である。ROM2はこの発明に関わるファクシミリ制御手順等のプログラムを記憶した読出専用メモリである。RAM3はこの発明に関わるファクシミリ制御のときに必要なデータを一時保管する読み書き可能なメモリである。

【0031】タイマ4はFAX文書の画像データの保持期間を管理する。画像蓄積メモリ5はFAX文書の画像データを蓄積保存するハードディスク装置、光ディスク装置等の記憶装置である。メール宛先管理テーブル6はFAX文書の画像データの配送先毎の管理テーブルである。サブアドレス・メールアドレス対応表7はFAX文書の画像データを電子メールで配送するクライアント端末装置の宛先を決定する参照テーブルである。

【0032】LANコントローラ8はネットワーク通信の制御を司る。トランス9はネットワーク上へのデータ転送を行なう。モデム10は2値符号化された画情報を公衆電話網に伝送可能に変調する処理、及び相手先からの変調された画像データの符号を元の2値符号に戻す復調処理を行なう。NCU11は公衆電話網とのファクシミリ通信を制御する。

【0033】符号化・復号化器12はスキャナ13で読み取った2値画像を圧縮して符号化する処理、及び相手先から送られた圧縮符号をプリント可能な2値画像に復



号化する処理を行なう。スキャナ13は原稿の画像を読み取る画像読取装置である。プリンタ14は復号化された画像データを印刷するレーザプリンタ、インクジェットプリンタ等の印刷装置である。

【0034】すなわち、上記CPU1等の機能部が、インターネット、イーサネット等のネットワークとの電子メールの通信制御を行なう手段と、公衆電話網との画像データのファクシミリ通信制御を行なう手段と、上記公衆電話網から画像データとその宛先のアドレス情報とを受信して上記画像データを蓄積する手段と、上記宛先のアドレス情報に基づいてその宛先に画像データの着信を知らせる着信通知メールを作成する手段と、その手段によって作成された着信通知メールを上記ネットワーク上のサーバへ送信する手段と、上記サーバを介して上記着信通知メールの宛先から画像要求メールを受信する手段と、その手段によって受信した画像要求メールに基づいて上記蓄積した中から該当する画像データの電子メールを作成する手段と、その手段によって作成された電子メールを上記サーバへ送信する手段の機能を果たす。

【0035】また、上記公衆電話網から受信した画像データの宛先毎に画像要求メールの着信の有無を管理する宛先管理テーブルと、その宛先管理テーブルに基づいて全宛先から画像要求メールを受信してそれぞれに画像データの電子メールを配信済みにしたと判断した後、上記蓄積した中から該当する画像データを削除する手段の機能を果たす。

【0036】さらに、上記画像データに設定された保持時間をカウントするタイマと、そのタイマによって保持時間が経過したとき、上記蓄積した中から該当する画像データを削除する手段の機能を果たす。

【0037】また、上記着信通知メールに画像データが削除される予定時間を付加する手段と、上記画像データに設定された保持時間をカウントするタイマと、そのタイマによって保持時間が経過したとき、上記画像要求メールの未受信の宛先へ画像データが削除されることを知らせる警告メールを作成して送信する手段と、その手段によって警告メールを送信した後にさらに所定時間を経過したとき、上記蓄積した中から該当する画像データを削除する手段の機能を果たす。

【0038】さらに、上記着信通知メールに画像データが削除される予定時間を付加する手段と、上記画像データに設定された保持時間をカウントするタイマと、上記タイマによって保持時間が経過したときに上記画像要求メールの未受信の宛先があったとき、上記蓄積した中から該当する画像データをプリント出力した後に削除する手段の機能を果たす。

【0039】また、上記画像データに設定された保持時間をカウントするタイマと、そのタイマによって保持時間が経過したとき、上記画像データに複数の個人宛が指定されていたとき、上記画像要求メールの未受信の宛先

に対して上記蓄積した中から該当する画像データを一定時間間隔でそれぞれ送信した後に上記該当する画像データを削除する手段の機能を果たす。

【0040】さらに、上記宛先から画像データの電子メールの送信不要を指定する電子メールを受信したとき、その宛先への電子メールの送信を実行せずに送信済みとして処理する手段の機能を果たす。

【0041】図2はこのFAX-MGにおける基本的なファクシミリ通信のプロトコルを示す図である。このFAX-MGは、同図に示すプロトコルに基づいて相手先とファクシミリ交信し、相手先からの画像データをファクシミリ受信する。

【0042】まず、送信側(TX)は受信側であるFAX-MG(RX)と接続すると、CNG信号を発信して自装置がファクシミリ端末装置であることを宣言する。一方、受信側(RX)は回線を接続したら、自装置がファクシミリ装置であり、受信状態であることを宣言するためにCED信号を発信する。そして、受信側(RX)は続けて自装置の持つ機能を送信側(TX)に知らせるためにNSF信号(非標準機能)とDIS信号(標準機能)を発信する。

【0043】送信側(TX)は受信側(RX)から受け取ったNSF信号とDIS信号の内容に基づいてFAX文書を送信する際に用いる機能を決定し、受信側(RX)にNSS信号(非標準機能)とDCS信号(標準機能)を発信して知らせる。その後、決定した機能から画像データの伝送に用いられるモデムスピードでTCF(トレーニング)を行なう。そのトレーニングが成功すると受信側(RX)はCFR信号を送信側(TX)へ返し、画像データの受信状態に移行する。

【0044】送信側(TX)は、CFR信号を受け取るとトレーニングで用いたモデムスピードで受信側(RX)へ画像データを送信し、その画像データの送信を終了すると画像データ送信の終了を受信側(RX)に通知するためにEOP信号を発行する。受信側(RX)は画像データを正常に受信できたら送信側(TX)へMCF信号を発信する。そして、DCN信号の発信によって回線断をそれぞれ行なう。

【0045】このFAX-MGは、上記プロトコル中のDIS信号のなからのサブアドレス等の情報を配信情報として利用する。そして、サブアドレスと配信先のメールアドレスとの対応表であるサブアドレス・メールアドレス対応表7に基づいて、送信側(TX)から受け取ったFAX文書の画像データをどのメールアドレス先へ配信すべきかを決定する。

【0046】図3はこのFAX-MGのこの発明に関わるファクシミリ制御に関わるソフトウェア構成を示す図である。このソフトウェアは、「FAXプロトコル」と「TCP/IP」の2つのプロトコルスタックを持つ。

【0047】その「FAXプロトコル」は、T.30で

規定されるファクシミリプロトコルを制御する。「モデム (Modem) 制御」は公衆電話網 (PSTN) での FAX データ通信に使用する。「PCR 制御」は T. 4 で規定されるファクシミリの画像圧縮と伸張を処理する制御を行なう。そして、「図情報管理」は FAX イメージ情報 (画像データ) などを管理して蓄積する。

【0048】一方、「TCP/IP」はインターネット、イーサネット等のネットワーク上の通信プロトコルである。「SMTP (simple mail transfer protocol)」は電子メールの転送プロトコルである。「MIME (multipurpose internet mail extension)」は、SMTP で扱うことができるデータがキャラクタデータであるのに対してバイナリデータをキャラクタデータに変換するための制御を行なう。「メール作成」は FAX 文書の画像データの電子メールを作成する。そして、「全体制御」はユーザインタフェース等のその他の処理を行なう。

【0049】図4はこの FAX-MG における FAX 文書を電子メールで配信する処理を示すデータフロー図である。この FAX-MG は、プロセス (図中「P」で示す) 1 で公衆電話網を介して FAX 文書のデータを FAX 受信し、そのうちの画像データ (イメージデータ) をメモリに蓄積する。

【0050】また、FAX 文書の受信時の FAX プロトコル受け取った転送情報を含むサブアドレス等のデータを電子メールの宛先を決定する P2 へ送り、「Sub adr.」「Mail adr.」との対応を示したサブアドレス・メールアドレス対応表を参照して電子メールの宛先を決定する。

【0051】さらに、P3 へ移行して P2 で決定された電子メールの宛先に基づいて FAX 文書の着信通知メールを作成し、P4 へ進んでその電子メールの適切な配信プロトコルによってネットワーク上のメールサーバへ着信通知メールを送信する。

【0052】その後、ネットワークを介して着信通知メールの宛先のクライアント端末装置からの FAX 文書の要求があれば、P5 でメモリにストアしている該当する画像データを MIME エンコードでキャラクタデータに変換し、P6 へ進んで FAX 文書の宛先に対応した電子メールを作成し、P4 によってその画像データ付きの電子メールを適切なプロトコルによってネットワーク上のメールサーバへ転送する。このようにして、FAX 文書を電子メールに変換してネットワーク上のクライアント端末装置へ配信する。

【0053】図5はこの FAX-MG を用いたネットワークシステムの一構成例を示す図である。このネットワークシステムは、図1に示した FAX-MG 20 が公衆電話網 21 を介して他のファクシミリ装置 (FAX) 22 とファクシミリ通信可能に接続されている。この公衆

電話網 21 を介して接続する FAX は複数台であるが同図では便宜上 1 台のみ示している。また、FAX-MG 20 はインターネット、イーサネット等のネットワーク 23 に対するネットワーク通信が可能に接続されている。

【0054】一方、ネットワーク 23 にはメールサーバ 24 と複数のクライアント端末装置 30, 31, 32, ……が接続されている。このメールサーバ 24 とクライアント端末装置 30, 31, 32, ……はそれぞれマイクロコンピュータを備えたパーソナルコンピュータ等の装置である。

【0055】メールサーバ 24 は、各クライアント端末装置 30, 31, 32, ……のメールボックス 26, 27, 28, ……を有するメモリ 25 を備えており、その各メールボックス 26, 27, 28, ……にネットワーク 23 を介して送られた各クライアント端末装置 30, 31, 32, ……宛の電子メールを蓄える。

【0056】図6は、図5のネットワークシステムにおける FAX 文書の配信処理を示すフロー図である。まず、FAX-MG 20 は FAX 22 から受信した FAX 文書を画像蓄積メモリ 5 に蓄積し、それを電子メールの形式にしてメールサーバ 24 へ転送する。その宛先は、FAX-MG 20 で管理するサブアドレス・メールアドレス対応表 7 と FAX 22 との交信で得られたサブアドレス等の情報を用いる。

【0057】メールサーバ 24 は、受け取った電子メールをその宛先に基づいて各メールボックス 26, 27, 28, ……にそれぞれコピーする。そして、各クライアント端末装置 30, 31, 32, ……はネットワーク 23 を介してポーリングによって自装置のメールボックスに電子メールが存在するか否かを確認し、存在していたときにはそれを取り込んで受け取る。こうして、FAX 22 からクライアント端末装置 30, 31, 32, ……へ FAX 文書を電子メールで転送することができる。

【0058】次に、FAX-MG 20 が複数の宛先に対する同じ内容の FAX 文書をファクシミリ受信したときの配信処理を説明する。図7はその処理の流れを示すフロー図、図8はサブアドレス・メールアドレス対応表のフォーマット例を示す図、図9は着信通知メールのヘッダ部分のフォーマット例を示す図、図10は着信通知メールのフォーマット例を示す図である。

【0059】図7に示すように、FAX 22 から FAX-MG 20 へ FAX 文書が送信されると、FAX-MG 20 はその FAX プロトコル上の DIS 信号内のサブアドレスとサブアドレス・メールアドレス対応表 7 とを参照して配信する宛先のクライアント端末装置を決定する。

【0060】FAX-MG 20 は宛先のクライアント端末装置を決定すると、FAX 文書の着信を知らせる着信通知メールを作成し、画像データの代わりにメールサー

10

1

20

30

40

50

バ24へ送信する。

【0061】図8に示すように、サブアドレス・メールアドレス対応表は、サブアドレスと配信先の宛先を示すメールアドレスと個人及びグループを示す種類との参照テーブルである。

【0062】図9に示すように、着信通知メールのヘッダ部分は、RFC822で規定された「Message-Id:」や「From:」等のフィールドが必要であり、その「Message-Id:」フィールドの部分にFAX文書のイメージを一意に決定するIDを規定する。

【0063】また、「From:」フィールドには電子メールアドレスである画像要求メールの宛先（例えば、「fax@xxxx.co.jp」）を指定し、且つ、その後の丸括弧“（）”内に相手先のFAX文書の差出人情報（例えば、TSI、RTI）を設定する。

【0064】その「From:」フィールドに差出人情報を書き込み、その後で鍵括弧“<>”内に画像要求メールの宛先を書き込むこともできる。そのようにすれば、クライアント端末装置は着信通知メールを受け取ったときに、クライアント端末装置の要求メールの差出人がFAX文書の発信者を知ることができ、且つその着信通知メールでFAX文書のイメージを一意に決定することができる。

【0065】図10に示すように、着信通知メールの本文部分には図中破線枠で囲んだ文章のように、FAX文書の着信を知らせる内容を記入する。

【0066】再び図7に戻り、クライアント端末装置は、着信通知メールを受け取ると、その着信通知メールを単純に返信（リプライ）することによってFAX-MG20から自装置宛のFAX文書を電子メールで受け取ることができる。

【0067】まず、クライアント端末装置は、着信通知メールをリプライするとき、「In-Reply-To:」フィールドを作成し、その中に返信する元になった着信通知メールの「Message-Id:」を入れる。その「Message-Id:」にFAX文書のイメージを一意に決定するIDが入っているので、それを受け取るようにする。

【0068】また、「In-Reply-To:」フィールドを追加しない場合、着信通知メールの「Subject:」フィールド内に「Message-Id」と同じIDを書き込んでおく。通常、電子メールを返信する場合は同じ「Subject:」を流用する（先頭に「Re:」等のフィールドが付加されることもある）ので、その「Subject:」からFAX文書のイメージのIDを得ることができる。

【0069】なお、「Message-Id:」と「Subject:」のフィールドのIDが異なっていれば、ユーザの指定によっていずれか一方を優先的に採用

するようにする。

【0070】そして、FAX-MG20は、クライアント端末装置からの画像要求メールを受け取ると、その「From:」フィールドに基づいてFAX文書の画像データの送り先を決定し、「In-Reply-To:」フィールドで決定されたFAX文書の画像データをメールサーバ24へ送出する。

【0071】このようにして、FAX-MG20はFAX文書を受信したとき、その宛先のクライアント端末装置に電子メールによる着信通知メールを送って着信を知らせて、そのクライアント端末装置からの画像要求メールを受け取ったときに該当するFAX文書の画像データを電子メールで転送するので、電子メールをネットワーク23上の各クライアント端末装置30、31、32、……に配信するためのメールサーバ24に画像データからなる大容量の電子メールを貯め込むことを防止することができ、メールサーバ24の負荷を大幅に軽減させることができる。

【0072】次に、FAX文書の画像要求メールは各クライアント端末装置から別々にくるので、FAX文書の画像データを削除する時期を管理するとよい。そこで、このFAX-MG20における画像蓄積メモリ5に蓄積した画像データを削除する処理について説明する。まず、FAX-MG20は、メール宛先管理テーブル6を1回のFAX文書の実受信毎に用意する。

【0073】図11はメール宛先管理テーブルのフォーマット例を示す図である。このメール宛先管理テーブル6は、サブアドレス・メールアドレス対応表7のファクシミリ受信したFAX文書のサブアドレスに対応する部分から配信先のメールアドレスに対応する部分をコピーし、それぞれにFAX文書の画像データのイメージを一意に決定するIDを付加し、さらに状態フィールドを追加して作成する。

【0074】そして、着信通知メールを送信した直後は、配信先のメールアドレスに該当する状態フィールドに未配信の情報を記録する。その後、クライアント端末装置から画像要求メールを受信すると、その「Message-Id:」フィールドに示されたFAX文書の画像データのIDと「From:」フィールドに示された配信先アドレスが共に一致している部分を宛先管理テーブルから検索する。

【0075】その検索された配信先のアドレス宛に該当するFAX文書の画像データを電子メールに変換して送信し、先に検索された部分の状態フィールドの未送信を配信済み書き替える。但し、見つからなかったときは何もしない。そして、FAX文書の画像データのIDの全ての状態フィールドが配信済みになったら、その画像データを画像蓄積メモリ5から削除する。

【0076】図12はFAX文書とサブアドレス・メールアドレス対応表とメール宛先管理テーブルの参照関係

を示す説明図である。まず、FAX文書を受信すると、そのFAXプロトコル情報の「サブアドレス」を取得し、ステップ(図中「S」で示す)1でその「サブアドレス」に基づいてサブアドレス・メールアドレス対応表のサブアドレス欄を参照し、該当する「サブアドレス」に対応する「配信先メールアドレス」を決定する。

【0077】さらに、ステップ2でその「配信先メールアドレス」とFAX文書の「FAXイメージID」とによってメール宛先管理テーブルを作成する。このメール宛先管理テーブルは1回のFAX受信毎に作成する。

【0078】その後、クライアント端末装置から画像要求メールを受信すると、メール宛先管理テーブルの「状態」フィールドに基づいて宛先毎の配信状況を確認する。そして、ステップ3で画像要求メールにあった着信通知メールのメッセージIDに基づいてFAXイメージIDを決定し、そのFAXイメージ本体の画像データを取得する。

【0079】このようにして、FAX文書を全ての宛先のクライアント端末装置へ配信したことを確認した後に削除するので、全宛先に配信済みの不要になった画像データを画像蓄積メモリ5からタイミング良く削除することができ、不要な画像データを貯め込むことなく、FAX-MG20のメモリ容量を有効に活用することができる。

【0080】次に、FAX-MG20でFAX文書を転送するメールアドレスがメールサーバ24で管理するグループアドレスや別名アドレスのときは、メール宛先管理テーブル6上のアドレスと画像要求メール内のアドレスとが一致なくなるので、全ての宛先に配信したか否かを判断することが難しくなる。そこで、このFAX-MG20における画像蓄積メモリ5に蓄積した画像データを予め設定した有効期限が経過したときに削除する処理について説明する。

【0081】この処理では、配信先のメールアドレスに着信通知メールを送信した直後に、メール宛先管理テーブル6の配信先のメールアドレスに該当する状態フィールドに未配信の情報を記録する。但し、サブアドレス・メールアドレス対応表7の種類フィールドにグループ(又は、エイリアス)という属性が記録されている場合、未配信を記録する代わりに「N/A(無効)」を記録する。

【0082】また、メール宛先管理テーブル6の作成時、タイマの属性を付加する。そのタイマの属性にはユーザが予め設定した受信したFAX文書の画像データの保持時間を記録する。その後、クライアント端末装置から画像要求メールを受信すると、その「Message-ID:」フィールドに示されたFAX文書の画像データのIDと「From:」フィールドに示された配信先アドレスが共に一致している部分を宛先管理テーブルから検索する。

【0083】その検索された配信先のアドレス宛に該当するFAX文書の画像データを電子メールに変換して送信し、先に検索された部分の状態フィールドの未送信を配信済みに書き替える。但し、見つからなかったときは何もしないし、状態フィールドにN/A(無効)が記録されているときにはそのままにする。

【0084】そして、FAX文書の画像データのIDの全ての状態フィールドが配信済みになったら、その画像データを画像蓄積メモリ5から削除する。しかし、状態フィールドにN/A(無効)が記録されているFAX文書についてはグループなので削除しない。

【0085】そこで、タイマ4はメール宛先管理テーブル6内の全てのFAX文書の保持時間を検索し、現在の時間と比較して経過しているものを見つけ出す。その結果保持時間が経過したFAX文書の状態フィールドを配信済みに書き替える。この場合、既にN/A(無効)が記録されていても配信済みに書き替える。

【0086】そして、FAX文書の画像データのIDの全てのフィールドが配信済みになったら、そのFAX文書の画像データを削除する。したがって、グループの場合、予め設定した保持時間内なら状態フィールドの属性はN/A(無効)のままだが、その保持時間を経過すると配信済みに書き替えて削除する。

【0087】このようにして、配信先が複数のクライアント端末装置を含むグループの場合には、所定の時間を経過した後にFAX文書の画像データを削除することができるので、配信すべき全クライアント端末装置を特定できない場合にいつまでもFAX文書の画像データを蓄積したまま削除できなくなるような状態を回避することができる。

【0088】次に、タイマ4によって保持時間が経過した画像データを一方的に削除してしまうと、うっかり画像要求メールを出さなかったクライアント端末装置のユーザがFAX文書を受け取れなくなると不便なので、画像データの削除前に画像要求メールを受信していないクライアント端末装置へ画像データを削除することを警告するメールを送信して注意を促すとよい。そこで、このFAX-MG20におけるクライアント端末装置へFAX文書の画像データの削除を警告する処理について説明する。

【0089】この処理では、FAX-MG20は着信通知メールを作成して送信するとき、タイマ4で指定された画像データの保持時間を本文内に記入して(図10に一点鎖線枠で示す)送信する。そして、その着信通知メールの送信後、タイマ4でメール宛先管理テーブル6の画像データの保持時間を検索し、現在の時間と比較して経過したものを見つける前に画像データを削除することを警告する警告メールを作成する。

【0090】その警告メールの宛先はメール宛先管理テーブル6から該当するものを得ることができ、ヘッダに

は「Message-ID」「Subject」にFAX文書の「FAXイメージID」を付加する。こうして、クライアント端末装置側では受け取った警告メールをそのまま返信することによって画像要求メールにすることができる。

【0091】このようにして、各クライアント端末装置のユーザは、自装置宛のFAX文書が削除される期限を容易に知ることができ、その期限内に必要なFAX文書を要求することができる。

【0092】次に、警告メールの宛先にグループの宛先も含む場合、そのグループの全てのメンバー（クライアント端末装置のユーザ）から画像要求メールを受け取ったか否かを判断することが難しくなる。そのため、うっかり画像要求メールを出さなかったクライアント端末装置のユーザがFAX文書を受け取れなくなるし、毎回FAX文書を受信する度に無用な警告メールを送信してしまう恐れがあるので、FAX文書を記録紙にプリントして残しておくとの良い。そこで、このFAX-MG20におけるFAX文書のプリント処理について説明する。

【0093】この処理では、FAX-MG20はFAX文書を受信時、DIS信号の中のサブアドレスから配信先アドレスを決定し、その配信先アドレスに該当するサブアドレス・メールアドレス対応表7の「種類」フィールドを検査して、その「種類」フィールドにグループと記入されていた場合、プリンタ14を起動して受信したFAX文書の画像データを受信中にプリントする。

【0094】こうして、FAX文書の画像データの削除後にクライアント端末装置から画像要求メールを受け取ったときでも、その内容を記載したドキュメントとして容易に提供することができる。

【0095】次に、配信先にグループやエイリアスが存在しないときでも、個人宛のアドレスのFAXを受信した場合、その宛先から画像要求メールがこないといつまでも画像データを削除できなくなり、画像蓄積メモリ5の領域を有効に活用することができなくなるので、所定の時間が経過したときに未だ画像要求メールが来ないクライアント端末装置へFAX文書の電子メールを送信すると良い。そこで、このFAX-MG20における所定時間経過後に未配信のクライアント端末装置へFAX文書の画像データを送信する処理について説明する。

【0096】この処理では、FAX-MG20はメール宛先管理テーブル6の作成時、タイマ4で管理する保持時間の他に送信予定時刻を追加して設定する。そして、通常は「状態」フィールドに「N/A（無効）」を設定する。

【0097】タイマ4によって保持時間切れのFAX文書の項目を見つけた場合、「状態」フィールドを「送信済み」を設定する代わりに「送信予定時刻」を有効値にする。その際、複数の宛先の送信予定時刻が見つかったら、各送信予定時刻を予め設定した一定時間ずつずらす

ように変更する。そして、その各送信予定時刻がきても未配信の場合、それぞれの宛先に対してFAX文書の画像データを電子メールにして配信する。

【0098】このようにして、複数の宛先へそれぞれ異なる時刻で未配信の電子メールを送信するので、メールサーバ24の負担を掛けずに済み、各クライアント端末装置のユーザに確実にFAX文書を配信することができる。

【0099】次に、クライアント端末装置のユーザが、例えばディスクスペースが足りない等の理由で画像要求メールを送信しなかった場合でも所定時間後にFAX文書の画像データが配信されてしまうとメールサーバ24に負担をかけてしまうので、クライアント端末装置からの通知に応じてFAX文書の配信と未配信を実行すると良い。そこで、このFAX-MG20におけるクライアント端末装置からの通知に応じてFAX文書の画像データの配信と未配信を切り換える処理について説明する。

【0100】この処理では、着信通知メールを受信したクライアント端末装置は、その着信通知メールに基づいて画像要求メール又は拒否メールを作成してFAX-MG20へ返信する。

【0101】図13は画像要求メールと拒否メールのヘッダ部分のフォーマット例を示す図、図14は画像要求メールと拒否メールのフォーマット例を示す図である。クライアント端末装置では、着信通知メールのヘッダ部分を用いて、図13に示すような画像要求メールと拒否メールのヘッダ部分を作成し、そのヘッダ部分を用いて画像要求メール又は拒否メールを作成する。そして、拒否メールのときには、図14に破線枠で示すように、本文に「No Use」のメッセージを記入する。

【0102】FAX-MG20は、クライアント端末装置からの電子メールを受信すると、その本文の先頭部分に画像データの送信を拒否するキーワードである「No Use」があるか否かを調べる。そのキーワードがなければ、画像要求メールと判断して該当するFAX文書の画像データを電子メールで配信する。

【0103】また、キーワードである「No Use」があったときには、拒否メールと判断してFAX文書の配信を行わずに、メール宛先管理テーブル6の「状態」フィールドを「配信済み」に書き替える。そして、そのFAX文書の「FAXイメージID」に該当する全ての「状態」フィールドが「配信済み」になったらFAX文書の画像データを削除する。

【0104】このようにして、クライアント端末装置のユーザの都合によってFAX文書の配信が必要無いときにも適切に応じることができる。

【0105】

【発明の効果】以上説明してきたように、この発明によるファクシミリ装置によれば、ネットワーク上のサーバに対して電子メール化した同じ内容のFAX文書を一度

に複数転送しなくても各クライアント端末装置が自装置宛の電子メールを滞り無く受け取ることができる。

【図面の簡単な説明】

【図1】この発明のファクシミリ装置の一実施形態であるファックスメールゲートウェイのハードウェア構成を示すブロック図である。

【図2】図1に示したFAX-MGにおける基本的なファクシミリ通信のプロトコルを示す図である。

【図3】図1に示したFAX-MGのこの発明に関わるファクシミリ制御に関わるソフトウェア構成を示す図である。

【図4】図1に示したFAX-MGにおけるFAX文書を電子メールで配信する処理を示すデータフロー図である。

【図5】図1に示したFAX-MGを用いたネットワークシステムの一構成例を示す図である。

【図6】図5に示したネットワークシステムにおけるFAX文書の配信処理を示すフロー図である。

【図7】図1に示したFAX-MGが複数の宛先に対する同じ内容のFAX文書をファクシミリ受信したときの配信処理の流れを示すフロー図である。

【図8】図1に示したサブアドレス・メールアドレス対応表のフォーマット例を示す図である。

【図9】図1に示したFAX-MGが送信する着信通知メールのヘッダ部分のフォーマット例を示す図である。

【図10】図1に示したFAX-MGが送信する着信通

知メールのフォーマット例を示す図である。

【図11】図1に示したメール宛先管理テーブルのフォーマット例を示す図である。

【図12】FAX文書とサブアドレス・メールアドレス対応表とメール宛先管理テーブルの参照関係を示す説明図である。

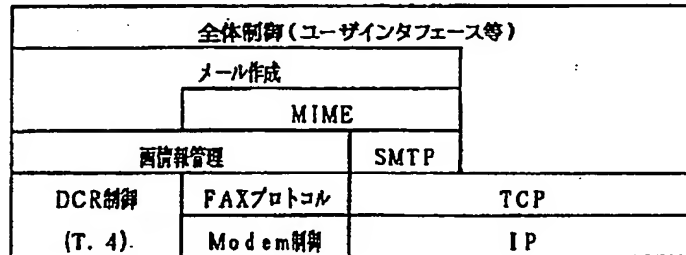
【図13】画像要求メールと拒否メールのヘッダ部分のフォーマット例を示す図である。

【図14】画像要求メールと拒否メールのフォーマット例を示す図である。

【符号の説明】

- 1 : CPU                      2 : ROM
- 3 : RAM                     4 : タイマ
- 5 : 画像蓄積メモリ    6 : メール宛先管理テーブル
- 7 : サブアドレス・メールアドレス対応表
- 8 : LANコントローラ    9 : トランス
- 10 : モデム (MODEM)
- 11 : ネットワークコントロールユニット (NCU)
- 12 : 符号化・復号化器 (DCR)
- 13 : スキャナ              14 : プリンタ
- 20 : ファックスメールゲートウェイ (FAX-MG)
- 21 : 公衆電話網
- 22 : ファクシミリ装置 (FAX)
- 23 : ネットワーク        24 : メールサーバ
- 25 : メモリ                26 ~ 28 : メールボックス
- 30 ~ 32 : クライアント端末装置

【図3】



【図8】

サブアドレス・メールアドレス対応表

ID	サブアドレス	配 信 先	種 別
1	0000	aaa@XXX.co.jp	個人
2		bbb@XXX.co.jp	個人
3		ccc@XXX.co.jp	個人
4	101	group@XXX.co.jp	グループ
5	20021	ddd@XXX.co.jp	個人
6		group@XXX.co.jp	グループ

【図9】

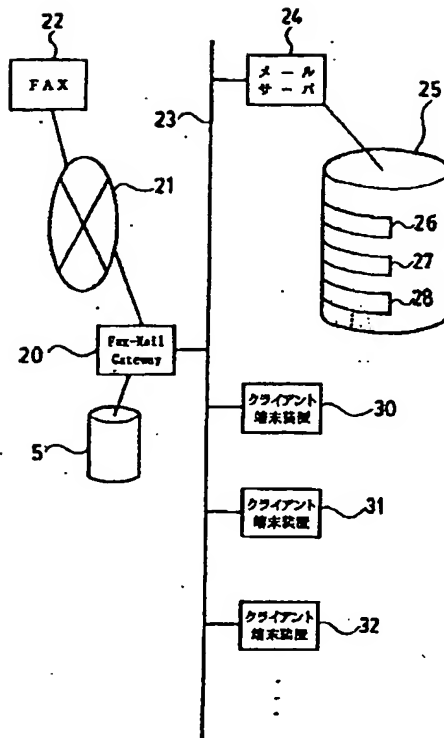
着信通知メールのヘッダ部分

```

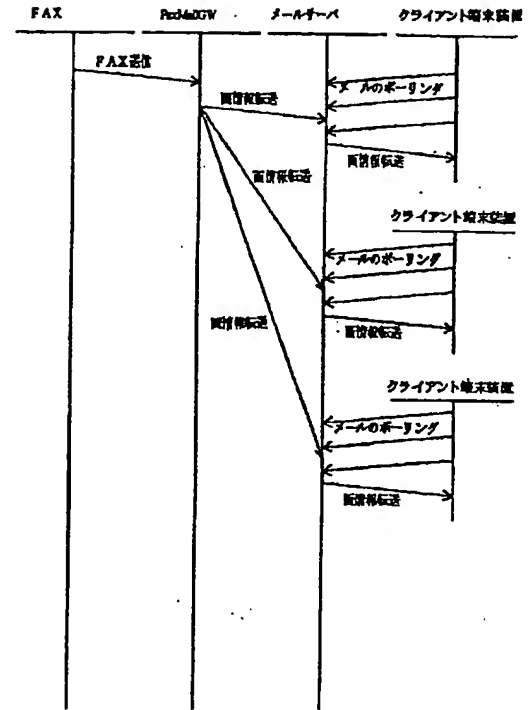
Message-Id: <906211500.1100563@XXX.co.jp>
Date: Sat, 22 Jun 1996 00:00:01 +0900
From: fax@XXX.co.jp(Fax message from 041224458)
To: ooon@XXX.co.jp
Subject: a Fax message 906211500.1100563
Reply-To: fax@XXX.co.jp

```

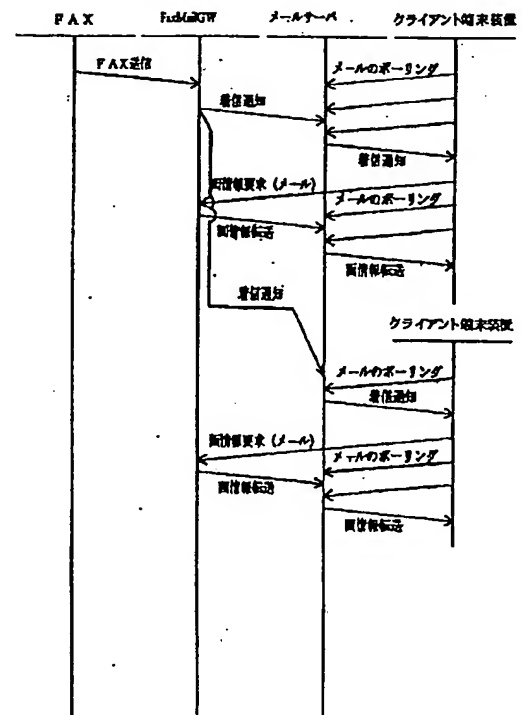
【図5】



【図6】



【図7】



【図13】

受信文書(FAXイメージ)要求メール及び経過メールのヘッダ部分

```

Message-Id: <9606211500.AA00563@IIX.co.jp>
Date: Sat, 21 Jun 1996 00:00:01 +0900
From: fax@IIX.co.jp
To: ooca@IIX.co.jp
Subject: Re: a Fax message 9606211500.AA00563
In-Reply-To: <9606211500.AA00563@IIX.co.jp>; from
fax@IIX.co.jp at Sat, 21 Jun 1996 00:00:01 +0900

```

【図14】

受信文書(FAXイメージ)経過メール

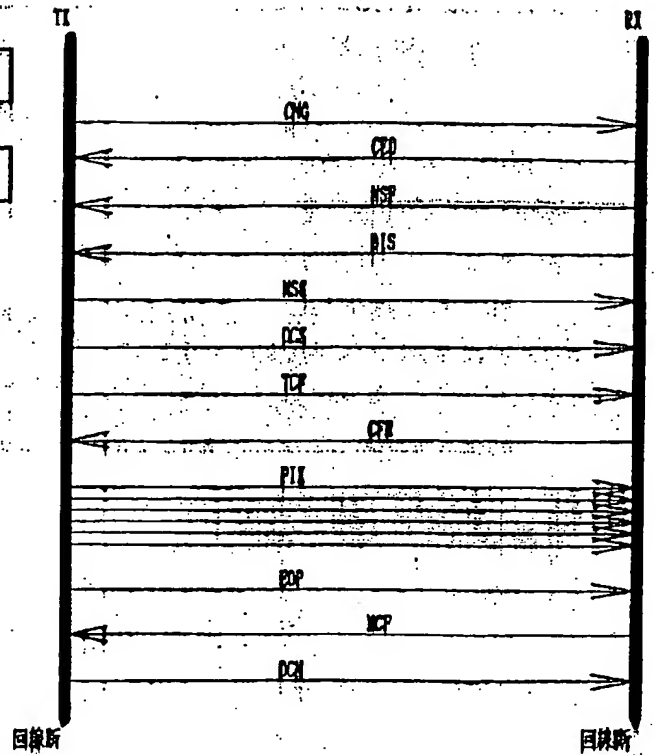
```

Message-Id: <9606211500.AA00563@IIX.co.jp>
Date: Sat, 21 Jun 1996 00:00:01 +0900
From: fax@IIX.co.jp (Fax message from 941222438)
To: ooca@IIX.co.jp
Subject: Re: a Fax message 9606211500.AA00563
In-Reply-To: <9606211500.AA00563@IIX.co.jp>; from
fax@IIX.co.jp at Sat, 21 Jun 1996 00:00:01 +0900

```

Bo Ueq

【图2】



【圖 10】



00000 ILL. CO. Jp 000

ご利用ありがとうございます。



【図11】

10	登録番号	PAX4-X-701D	検査	登録	22.15:00:00
1.1	aaa@111.co.jp	8903211500.1A07523	送付待ち	送付待ち	22.15:00:00
1.2	bbb@111.co.jp	8903211500.1A07523	送付待ち	送付待ち	22.15:00:00
1.3	ccc@111.co.jp	8903211500.1A07523	送付待ち	送付待ち	22.15:00:00
1.4	ddd@111.co.jp	8903211522.1A07543	送付待ち	送付待ち	22.15:23:00
1.5	eee@111.co.jp	8903211522.1A07543	N/A	N/A	22.15:23:00
1.6	fff@111.co.jp	8903211522.1A07543	N/A	N/A	22.15:23:00
1.7	ggg@111.co.jp	8903211522.1A07543	送付待ち	送付待ち	22.15:23:00
1.8	hhh@111.co.jp	8903211522.1A07543	送付待ち	送付待ち	22.15:23:00
1.9	iii@111.co.jp	8903211522.1A07543	送付待ち	送付待ち	22.15:23:00
1.10	jjj@111.co.jp	8903211522.1A07543	N/A	N/A	22.15:23:00

X-M 宛先管理テーブル

【図12】

